



**UNIVERSIDAD DE LAS CIENCIAS INFORMATICAS  
CALIDAD DE SOFTWARE FACULTAD 7**

***Título: ¿Cómo gestionar la configuración para lograr  
un producto con calidad en la UCI?***



**ModelConfig.UCI**

**Trabajo de Diploma para optar por el título de  
Ingeniero en Ciencias Informáticas**

**Autoras:**

Yaritza Magadán Labrada

Yeniset León Perdomo

**Tutora:**

Ing. Irina Brito Reyes

**Cotutora:**

Ing. Lizandra Arza Pérez

**Consultante:**

Ing. Ramsés Delgado Martínez

**Julio 2007**



*“(...) tenemos que producir no solamente más, sino mejor. Mejor en cuanto a calidad, a durabilidad y al uso efectivo del producto, y mejor en cuanto a la presentación, en cuanto a la capacidad de exportación en algunos casos...”*

*Che*

## **Declaración de Autoría:**

Declaramos que somos las únicas autoras de este trabajo y autorizamos a la Dirección de Calidad de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los 21 días del mes de junio del año 2007.

Yeniset León Perdomo

\_\_\_\_\_  
Nombre completo del primer autor

Yaritza Magadán Labrada

\_\_\_\_\_  
Nombre completo del segundo autor

Ing. Irina Brito Reyes

\_\_\_\_\_  
Nombre completo del primer tutor

Ing. Lizandra Arza Pérez

\_\_\_\_\_  
Nombre completo del segundo tutor

## *Dedicatoria*

*A mi mamá, a mi papá y hermano.*

*A mis abuelos, en especial a mi abuela*

*Concha.*

*Yary*

*A mi mamá, a mi papá y a mi hermano.*

*A mis abuelos, en especial a mi abuelo Cora.*

*A toda mi familia.*

*Yeni*



## **Agradecimientos**

A nuestro Comandante en Jefe Fidel Castro por brindarnos la oportunidad de estudiar en esta grandiosa universidad, y a la Revolución Cubana por darnos el derecho a todos a tener una educación gratuita y de una alta calidad.

A nuestra tutora Irina, por todos los consejos y la ayuda que nos ha brindado, y por las horas que empleó en revisar nuestras tantas versiones. Gracias por ser tan exigente. Nos sentimos muy orgullosas de haberte tenido como tutora.

A Liza por ser nuestra cotutora y amiga. Gracias por apoyarnos y confiar en nosotras desde el principio. Eres excepcional.

A Ramsés por tener tanta paciencia con nosotras y atendernos cada vez que lo buscamos.

A la profe Alina, por su ayuda y apoyo cada vez que la necesitamos.

A Yannier por ayudarnos en la confección del logo del modelo. Nene te quedo lindísimo.

A Lourdes por compartir conocimiento y experiencia en la elaboración de la presentación.

A todos los muchachones del proyecto CALISOFT. Tienen todo nuestro cariño y respeto.

A las compañeras de apartamento. Gracias por las lindas experiencias compartidas, las largas discusiones, en fin, todo. Muchachitas son lindísimas...

A Tay por ayudarnos y preocuparse por la tesis. Gracias amiga

A Ailyn Febles, gracias por haber hecho tu tesis de doctorado.

A la profe Pura. Un millón de gracias por revisarnos el documento.

A nuestra decana por haber sido un ejemplo a seguir en estos 5 años.



A las direcciones de Recreación, Extensión y Protocolo...amigos, fueron imprescindibles, sin ustedes no lo hubiésemos logrado.

A la UCI por ser nuestra segunda casa durante estos cinco años.

A la FEU y la UJC por quitarnos tanto tiempo y contribuir a nuestra preparación integral.

A todos aquellos que en algún momento nos preguntaron ¿cómo va la tesis?

A todos nuestros amigos...



*Yeni*

A mami y papi por darme todo su amor y cariño. Gracias por sus desvelos. Gracias por creer en mí.

A mi hermano por tenerme como ejemplo y a mi cuñada por entrarlo en caja.

A papá Cora, mamá Mulata y mamá Cuca, gracias por quererme tanto, gracias por tenerme como su niñita...

A abuela Tina y mamá Lila por tenerme presente a pesar de sus achaques.

A mis tíos (Arbelio, Jesús, Omar, Hugo) y a mis tías (Maidel, Luisa, Petra, China, Mary, Chiqui) por brindarme siempre su apoyo y preocuparse tanto por mí.

A todos mis primos (Yane, Arbelito, Anye, Yomarcito, Yosmany, Yoandy, y a los Yase) por quererme mucho.

A mi pequeño sobrino Andy Jesús. Te quiero un mundo.

*Yary*

A mis padres por todo el amor y el apoyo que me han brindado siempre.

A mi hermano por las broncas y el cariño que siempre me has dado. Te deseo lo mejor del mundo.

A mis abuelos Cira, Pepe, mamama y papapa, por su amor y apoyo financiero.

A Cheni y mi primo Narcy por su aprecio y regalos...

A mis tíos, en especial Hectorín, Muñi, Yoanny y Darío. Los quiero mucho.

A mis tías Alina, Bárbara, Yosde y Raque, por su apoyo y por compartir buenos momentos que nunca olvidaré. Son parte de mi vida.

A Xiomara, Chichi y Tinito por acogerme en su seno familiar y hacerme parte de ella. Gracias por todo.

A toda mi familia de la Habana por darme cobijo estos cinco años, gracias por apoyarme tanto.



A Day por hacerme reír tanto con sus pinareñadas, gracias amiga por ser como eres.

A Liza, sobran las palabras...

A mi tata por comprenderme, aguantarme, y estar siempre a mi lado.

A mi tía Mary por ser amiga y hermana. Gracias por hacerme sentir parte de tu familia.

A Yosbel por ser mi amigo en las buenas y las malas.

A todos mis profesores amigos, Irina, Serguei, Yude, por compartir tantos momentos agradables y sacarme de la rutina.

A mis vecinos por preguntar por mí.

A la FEU por ser mi segunda escuela y permitirme vivir tan buenos momentos.

A Yary por ser la mejor compañera de trabajo que he tenido y por todos los desvelos que hemos pasado juntas. Gracias por organizar un poco mi vida y abrirme los ojos.

A todos mis amigos...

A mis primas Yari, Isa y Day. Siempre fueron un ejemplo para mí.

A Javier por quererme, comprenderme y aguantarme. Eres de las mejores cosas que me ha pasado en la vida.

A mis vecinos en especial a Masita y Arael. Un beso para todos.

A todos mis primos. Son muchos...los quiero demasiado.

A mis profes amigos, Yuse, Ariel y Nora por los primeros y quererme tanto

A todos mis primos. Son muchos...los quiero demasiado.

A Pablito por considerarme su hija. Cariños para ti.

A mi compañera de tesis por su paciencia, invitaciones, conocimientos y las trasnochaderas. Éxitos en tu vida profesional, futura vicerrectora.

A todos mis amigos en especial el piquete de la farándula. Que aguante!!!!



## **Resumen**

La Industria Cubana del Software (InCuSoft) está llamada a constituir un renglón de enorme importancia para la economía del país en un plazo de tiempo visiblemente corto, sin embargo las ideas de convertir nuestros servicios informáticos en un rumbo exportable se ven frenadas por diversos problemas que atentan contra la calidad del proceso productivo, y por consiguiente, contra los resultados del producto final. Muchos de estos problemas tienen su raíz en malas prácticas de ingeniería y gestión de software durante el proceso de desarrollo, que pueden ser solucionados a partir de la puesta en práctica de la disciplina conocida como Gestión de Configuración de Software (GCS).

En el presente trabajo se realiza un análisis de cómo se efectúa la disciplina de GCS en los diferentes proyectos productivos de la Universidad de las Ciencias Informáticas (UCI), obteniendo las buenas prácticas de los mismos para su posterior utilización. Como resultado, se obtiene el diseño de un Modelo de Referencia para la GCS en la UCI ModelConfig.UCI, efectuando un enriquecimiento en su definición. Haciendo así de este, un instrumento mucho más completo en el campo de la GCS, el cual, utilizado propiamente aumenta la calidad de los productos en el desempeño de la industria del software desde sus actividades más básicas hasta las de mayor complejidad, creando una disciplina en el proyecto desde estudiantes hasta profesores y directivos, que se revierte directamente en el aumento del dinamismo, adaptabilidad y eficiencia necesarios en cualquier proyecto de desarrollo de software.



## Índice

Capítulo I: Fundamentación Teórica .....	7
1.1 Estado del arte de la industria del software en el mundo. ....	8
1.1.1 La Industria de Software en Latinoamérica .....	10
1.1.2 Las tres “I” en el desarrollo de Software.....	11
1.2 Estado actual de la Industria Cubana de Software .....	12
1.3 Estado actual de la Gestión de Configuración en la Universidad.....	15
1.4 Gestión de Configuración de Software. Definiciones .....	19
1.5 Gestión de Configuración de Software. Valoraciones.....	20
1.6 Modelos y estándares tenidos en cuenta en la investigación .....	21
1.6.1 Gestión de Configuración del Software en los Modelos de Calidad.....	21
1.6.1.1 IEEE (Instituto de Ingenieros Eléctricos y Electrónicos).....	22
1.6.1.2 CMMI Integración del Modelo de Madurez de las Capacidades. ....	22
1.6.1.3 ISO. Organización Internacional de Normalización. ....	25
1.6.2 Investigación sobre los modelos existentes a nivel mundial.....	26
1.6.3 La vinculación de estándares.....	27
1.6.4 Un proceso para el desarrollo de software (RUP) .....	28
Conclusiones parciales .....	29
Capítulo II: La Gestión de Configuración en la UCI. ....	30
2.1.1 La entrevista.....	31
2.1.2 La Revisión de bibliografía .....	31
2.1.3 La encuesta.....	31
2.2 Etapas en la elaboración de una encuesta. ....	31
2.2.1 Etapa I: Formulación de objetivos. ....	32
2.2.2 Etapa II: Diseño de la muestra.....	33
2.2.2.1 Notación .....	33
2.2.3 Etapa III: Confección del cuestionario. ....	37
2.2.3.1 Clasificación de variables .....	37
2.2.3.2 Estructura de un Cuestionario Tipo .....	38
2.2.4 Etapa IV: Trabajo de campo (recolección de los datos) .....	38



---

2.2.5 Etapa V: Tareas posteriores a la recolección de datos. ....	39
2.2.6 Etapa VI: Procesamiento, análisis y presentación de los datos. ....	39
2.3 Conclusiones arrojadas según el procesamiento de encuestas. Análisis de algunas preguntas significativas. ....	50
2.4 Ejemplos prácticos de la Gestión de Configuración en la UCI. ....	54
Conclusiones parciales .....	57
Capítulo III: Modelo de Referencia para la Gestión de Configuración (ModelConfig.UCI).....	58
3.1 Análisis de la Gestión de Configuración en cuatro proyectos productivos. ....	58
3.2 Modelo conceptual para la definición de ModelConfig.UCI.....	59
3.3 Sistema de procesos incluido en ModelConfig.UCI.....	61
3.3.1 Procedimientos y gráficos en el Sistema de procesos.....	67
3.3.2 La identificación de los elementos de configuración .....	67
3.3.2.1 Consideraciones generales para el proceso .....	68
3.3.2.2 Procedimiento para la identificación de los Elementos de Configuración del Software.....	69
3.3.3 Los cambios a los elementos de configuración .....	71
3.3.3.1 Consideraciones generales para el proceso .....	72
3.3.3.2 Procedimiento para la gestión de los cambios.....	73
3.3.4 La planificación en la gestión de la configuración.....	76
3.3.5 Las versiones en la gestión de configuración.....	76
3.3.6 Relación del Sistema de procesos de ModelConfig.UCI con RUP.....	79
3.4 Sistema de métricas en ModelConfig.UCI.....	80
3.4.1 Métricas de la gestión de los cambios y la planificación .....	82
3.4.1.1 Métrica para el control del avance del proyecto en la ejecución de los cambios (AvPy).....	83
3.4.1.3 Métrica para medir el cumplimiento de lo planificado (CPI).....	84
3.4.1.4 Métrica para medir el esfuerzo de los miembros del equipo (EfP) .....	85
3.4.2 Métricas que vinculan la Identificación de los Elementos de Configuración y el Control de Versiones .....	86



---

3.4.2.1 Métrica: Elementos de Configuración del Software en cada versión de un proyecto. (ECpV) .....	87
3.4.2.2 Métrica: Estado de los elementos de configuración (EEC).....	87
3.4.2.3 Métrica: Versiones de cada Elementos de Configuración del Software registradas en la Línea Base (VLB) .....	88
3.4.3 Métricas que vinculan la Gestión de los Cambios y el Control de Versiones .....	89
3.4.3.1 Métrica: Pedidos de cambio asociadas a una versión de un producto (PcV).....	90
3.4.3.2 Métrica: Versiones libres de defectos en una etapa (VLD).....	91
3.5 Sistema Informático de ModelConfig.UCI .....	91
3.5.1 Herramientas para la Gestión de Configuración .....	93
3.5.1.1 IBM Rational ClearQuest .....	94
3.5.1.2 Rational ClearCase.....	94
3.5.1.3 Microsoft Visual SourceSafe. VSS.....	95
3.5.1.4 Subversion .....	96
3.5.1.5 Source Forge .....	96
3.5.2 ConfigCASE 3.0, herramienta CASE para le control de configuración .....	97
3.5.2.1 Antecedentes.....	97
3.5.2.2 Características generales de ConfigCASE 3.0 .....	98
3.5.3 Comparación de otros productos con el propuesto.....	99
3.5.4 Consideraciones para el desarrollo de la herramienta.....	100
Conclusiones parciales .....	104
Conclusiones .....	106
Recomendaciones .....	109



## Índice de Tablas

Tabla 1.1 Datos generales de la industria del software en algunos países de Latinoamérica.....	10
Tabla 1.2 Organizaciones de Israel, Irlanda y La India.....	12
Tabla 2.1 Etapas para el diseño y la elaboración de una encuesta.....	32
Tabla 2.2 Tamaños de muestra por estratos de la población.....	35
Tabla 2.3 Ventajas comparativas para cada metodología.....	39
Tabla 2.4: Por ciento de los estudiantes por proyectos productivos.....	40
Tabla 2.5 ¿Se dedican solo a la producción de software?.....	42
Tabla 2.6 Otras actividades que se realizan en los proyectos productivos.....	42
Tabla 2.7 Cantidad de productos fabricados.....	42
Tabla 2.9 Calidad de los productos que se fabrican en la UCI.....	44
Tabla 2.10 Evaluación de los estudiantes y profesores según su definición del concepto de calidad.....	45
Tabla 2.11 Equipo de aseguramiento de la calidad por proyectos.....	45
Tabla 2.12 Personal dedicado a la Gestión de Configuración por proyectos.....	45
Tabla 2.13 Proyectos que poseen el rol de Administrador de Configuración.....	47
Tabla 2.14 Plan de Gestión de Configuración aplicado en cada proyecto.....	47
Tabla 2.15 Identificación y gestión de las versiones existentes de los programas y la documentación por proyecto.....	47
Tabla 2.16 Control de los cambios antes y después de que el software sea distribuido al cliente.....	48
Tabla 2.17 Logros que se obtienen con la disciplina de Gestión de Configuración.....	48
Tabla 2.18 Importancia de la Gestión de Configuración según cada proyecto.....	49
Tabla 2.19 Disciplina de Gestión de Configuración en las facultades.....	50
Tabla 3.1 Relación RUP-ModelConfig.UCI.....	80
Tabla 3.2 Comparación entre herramientas CASE.....	100



**Índice de figuras**

Figura 1.1 Esquema de la investigación. ....	8
Figura 1.2: Proceso de Calidad en la UCI. Estructura .....	17
Figura 1.3 Plan de evaluación del Software.....	18
Figura 1.4 Vinculación de ISO y CMM.....	28
Figura 2.1 Clasificación de variables .....	37
Figura 2.2 Calidad de los productos que se fabrican en la UCI .....	51
Figura 2.3 Evaluación de los estudiantes y profesores según su definición del concepto de calidad .....	52
Figura 2.4 Equipo de aseguramiento de la calidad por proyectos .....	52
Figura 2.5 Personal dedicado a la Gestión de Configuración por proyectos .....	53
Figura 2.6 Disciplina de Gestión de Configuración en las facultades .....	53
Figura 3.1 Modelo conceptual de ModelConfig.UCI .....	60
Figura 3.3 Gráfico del proceso de Identificación de los Elementos de Configuración .....	71
Figura 3.4 Diagrama de actividad # 1 del proceso Gestión de Cambios en la Configuración .....	74
Figura 3.5 Diagrama de actividad # 2 del proceso Gestión de Cambios en la Configuración .....	75
Figura 3.6 Diagrama de actividad # 1 del proceso Gestión de Versiones .....	78
Figura 3.7 Diagrama de actividad # 2 del proceso Gestión de Versiones .....	79
Figura 3.8 Sistema Informático de ModelConfig.UCI .....	92



## **Introducción**

La industria del software en el mundo se ha desarrollado notablemente en los últimos años, sin embargo, los resultados alcanzados no cubren las expectativas iniciales vislumbradas. Esto se debe, básicamente, a que la productividad que se alcanza de manera general, es baja, la cantidad de recursos a consumir -en tiempo - es alta y el trabajo realizado casi nunca tiene la calidad requerida. Los proyectos se concluyen en fecha posterior a lo planificado y los problemas no se detectan a tiempo en estos medios indisciplinados y caóticos de desarrollo. [Lage, 2000]

Desde hace algunos años, vinculado al desarrollo de la InCuSoft en la UCI, se viene desarrollando un grupo de investigaciones relacionadas con las áreas de procesos del desarrollo del software, la madurez de las organizaciones, la calidad en los procesos y en el producto final. [Febles, 2000a]

En los diagnósticos realizados en estas investigaciones, desarrollado en coordinación con la Dirección de Calidad de la UCI, se han detectado diversos problemas, dentro de los que se destacan [Febles, 2000a]:

### **Planificación y Seguimiento**

No existe una adecuada planificación del trabajo.

No se puede dar seguimiento al progreso del proyecto (no hay definición de tareas, relaciones entre estas, cronogramas, puntos de chequeo, etc.).

Falta de objetividad en la apreciación de la evolución del trabajo de los especialistas al no poseer los mecanismos para detectar la magnitud del trabajo desarrollado.

Los proyectos concluyen, casi siempre, fuera de fecha.



## Mediciones

- No se puede comparar un proyecto nuevo con los anteriores y aprender de éstos.
- No se miden los procesos, ni los productos con métricas rigurosas lo que implica que no se pueda controlar ni mejorar el proceso a partir de los resultados en su ejecución.

## Organizativas

Pocos proyectos tienen definidos sus procesos a partir de los principios de la ingeniería y la gestión de software.

Es pobre la comunicación efectiva entre los involucrados en la producción.

Este trabajo de diploma se encargará de explicar la urgencia de garantizar la disciplina en la ejecución de los procesos asociados a la Gestión de Configuración de Software. Además de demostrar la necesidad de hacer más eficaz, eficiente y efectiva la producción de software en la InCuSoft, en particular en la UCI, para minimizar las deficiencias apuntadas.

Este grupo de problemas implica a que se presente la siguiente **situación problemática** en la UCI:

El proceso de producción de software en la UCI se basa fundamentalmente en el desarrollo de los proyectos. En la actualidad, la calidad del producto está por debajo de las metas que se quieren alcanzar. Para lograr la excelencia es necesario crear y aplicar un modelo de Gestión de Configuración, desde que comienza el ciclo de vida del software hasta que concluye, permitiendo una mejor planificación y evaluación del trabajo.

La falta de este modelo implica que no exista la posibilidad de poder reutilizar código, ni componentes de software de manera eficiente y mantenimiento del producto. No se puede contar la mayoría de las veces con la versión estable de un fichero determinado y se incurre en errores o la pérdida de versiones de ficheros, producto del trabajo solapado de los desarrolladores. Además de no poder contar con una herramienta que automatice todos los procesos que están definidos.



Toda esta situación nos lleva a definir el siguiente **problema de la investigación**:

¿Cómo gestionar la configuración para lograr las producciones de software con calidad en la UCI?

Como consecuencia se plantea como **objetivo de la investigación**:

Proponer un modelo de Gestión de Configuración para lograr un producto con calidad en la UCI.

A partir del análisis del objetivo de la investigación se derivan los siguientes **Objetivos específicos**:

- Realizar un estudio del estado actual de la Gestión de Configuración en el mundo.
- Caracterizar la Gestión de Configuración en la UCI.
- Obtener un Sistema de Proceso, un Sistema de Métricas y un Sistema Informático.

Por consiguiente se determina que el **objeto de estudio** es el proceso de producción de software en la UCI y como **campo de acción**, la Gestión de Configuración en la producción de software.

Se presenta la siguiente **idea a defender**: El desarrollo de un modelo para la Gestión de Configuración aplicado al desarrollo de software en la UCI, garantizará que se obtengan los productos con la calidad requerida.

De aquí se establecen las siguientes **variables**:

**Independiente**: Modelo de Gestión de Configuración.

**Dependiente**: Producciones de software con calidad en la UCI.

Para lograr los objetivos trazados se acometieron las siguientes **tareas**:

- Analizar la bibliografía existente sobre el tema.
- Valorar los parámetros y principios de calidad de un software que rigen la Gestión de Configuración.



- Valorar las propuestas de Gestión de Configuración acorde con las condiciones de los software producidos en la UCI.
- Definir los procesos para la Gestión de Configuración de Software que formarán parte del modelo de referencia.
- Definir un esquema general del modelo de referencia para la Gestión de Configuración de Software.
- Definir instrumentos de control y métricas que sirvan para obtener mejores resultados a partir de los datos almacenados.

Entre los Métodos de trabajo científico utilizados se destacan:

Los métodos empíricos, **la entrevista** a estudiantes, especialistas en la rama y profesores que trabajan en la UCI con experiencias en el tema de la calidad de software, con el objetivo de indagar e intercambiar criterios para que a través de sus conocimientos adquiridos se pueda elaborar una guía que permita gestionar la configuración para un producto con calidad, teniendo como propósito principal que esta sea cada vez mayor. **La encuesta** a estudiantes y profesores que forman parte del proyecto de calidad, tanto de la universidad como de las facultades, teniendo en cuenta sus experiencias en la evaluación de productos, para concluir cual es el modelo más eficiente y más usado para mantener la calidad, y así de esta forma crear nuevas propuestas de gestión.

**Los métodos teóricos** también van a formar parte en la realización de esta tesis, ya que tienen gran importancia en el proceso de la investigación, utilizándose en la construcción y desarrollo de la teoría científica y en el enfoque general para abordar los problemas de la ciencia. En este caso, se utiliza **el análisis y la síntesis**, para lograr una mejor comprensión del fenómeno, dividiendo en partes su solución e integrándolas para obtener un resultado final, ya que esta es una de las formas más eficientes para lograr entender un problema con todas sus interrogantes. **La inducción y deducción** como razonamiento para permitir dar solución al problema y así pasar del conocimiento general a otros de menor nivel de generalidad. **El análisis histórico-lógico** para facilitar el estudio de la trayectoria real de los fenómenos y acontecimientos en el curso de la historia, investigando las leyes generales del funcionamiento y desarrollo de los mismos. Lo que permitirá conocer cómo se gestiona la configuración en estos momentos



y poder desarrollar una gestión más completa, constituyendo este el objetivo más importante en este trabajo.

La **Novedad Científica** de la Investigación se expresa en sus aportes fundamentales que son los siguientes:

- Concepción de un Modelo de Referencia original para introducir la Gestión de Configuración de Software en la UCI.
- La integración de procesos definidos, métricas formalizadas para constituir un modelo unificado, coherente y consistente.
- La integración de herramientas automatizadas en un sistema que facilita la introducción del modelo en la UCI.

**La Significación Práctica del trabajo** está en que permitirá disponer de un conjunto de procedimientos para la GCS, en un modelo de referencia, adaptado a las características de la UCI permitirá dar seguimiento al desarrollo de los productos de software, controlar la calidad de estos procesos y mejorarlos garantizando de esta forma una mayor calidad del producto final. Contar con métricas para medir el desarrollo de los proyectos permitirá mejorar de forma incremental los procesos, poder tomar acciones correctivas por parte de los directivos y los líderes de proyecto llevando a escalas superiores los resultados internos de los equipos de desarrollo.

Poder aplicar estos procedimientos de forma automatizada hará su aplicación más fácil y cómoda para los desarrolladores garantizando alcanzar más rápidamente una disciplina de trabajo. Estos resultados tendrán amplia utilización para la InCuSoft, y en particular para la UCI.

La tesis quedó estructurada en tres capítulos. El Capítulo I, referido al marco teórico y referencial de la investigación donde se realiza un análisis crítico del estado del arte en el tema de GC. El Capítulo II, contiene el estado actual y problemas de la disciplina de GC en la UCI, obtenido mediante la realización de una encuesta y una entrevista, donde se describen ejemplos prácticos de cómo se aplica la misma en



los principales proyectos productivos. El Capítulo III, se presenta el esquema general del modelo de referencia, se detalla el Sistema de Procesos y el Sistema de Métricas.



## Capítulo I: Fundamentación Teórica

En la economía moderna, la industria del software es la piedra angular de toda la innovación tecnológica, ya que impacta en forma directa o indirecta sobre todas las actividades económicas. La demanda por los productos del software y los servicios de la información tecnológicas tienen una de las tasas de crecimiento mundial más alta de la actualidad.

Este desarrollo globalizado se originó en países altamente desarrollados, pero también benefició sustancialmente a otros países como la India, Israel e Irlanda que bajo diferentes enfoques cuentan con una industria del software creciente.

El software como motor de la tecnología de información se introdujo en todos los ámbitos económicos, modificando la configuración de los productos y servicios ofrecidos. En el ámbito de los productos del software; hoy es posible acercar las brechas que hay entre los países desarrollados y aquellos en vías del desarrollo. Esto está definido porque la industria del software descansa su poder en la capacidad intelectual de las personas. [Ministerio de Economía y Producción República Argentina, 2004]

El esquema de la Fig. 1.1 representa los aspectos fundamentales del objeto de estudio teórico, que se centran con mayor prioridad en:

- Estado del arte de la industria de software a nivel internacional, en la InCuSoft, y en particular en la UCI.
- La GC, proceso considerado como la columna vertebral de la producción de software. [Preesman, 2000]

Caracterización de la GC en el mundo y diagnóstico del grado de aplicación en Cuba.

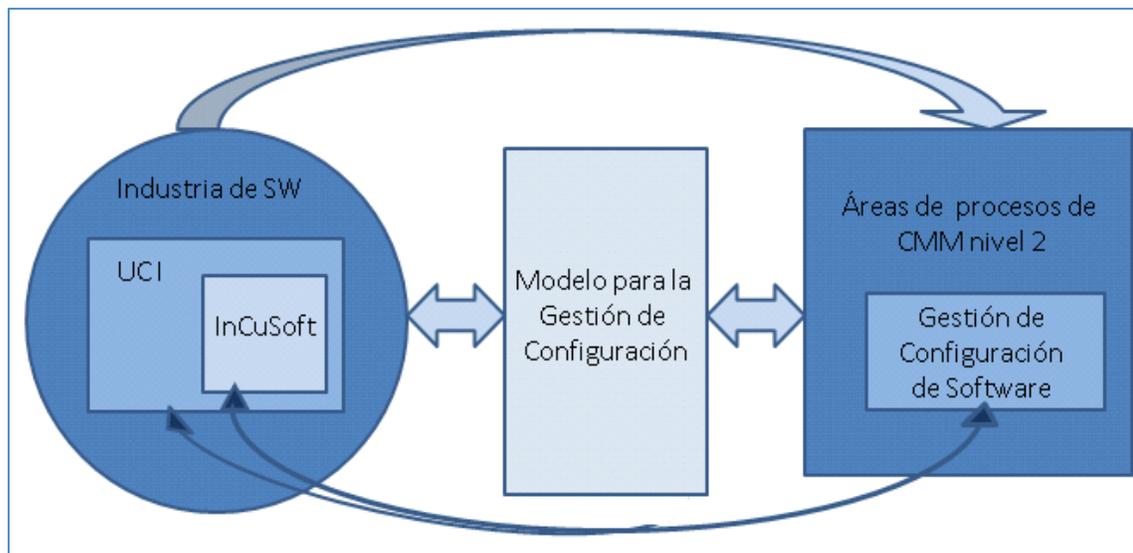


Figura 1.1 Esquema de la investigación.

### 1.1 Estado del arte de la industria del software en el mundo.

La innovación tecnológica, velocidad en el desarrollo y la satisfacción del cliente se han convertido en la consigna de las organizaciones que quieren sobrevivir y cada vez ser más competitivas en el mundo. [AMCIS, 2001]. Es esta una de las razones por las que el desarrollo de la industria del software ha pasado a ser prioridad de los gobiernos en muchos países de la región.

A pesar de este interés gubernamental y del indiscutible desarrollo de la industria del software en los últimos años, las estadísticas internacionales no son alentadoras. Informes de instituciones dedicadas al análisis de software muestran los siguientes indicadores [Reo, 2002] [Bedini, 1995] [Butler, 2001] [Camou, 2002]:

- El 25% de los proyectos de software son abortados.

Se liberan productos a sus clientes, con remanentes del 15% de defectos.



Muchas empresas gastan de 30% a 44% de su tiempo y dinero en trabajos sobre software ya liberado.

Se cumplen las planificaciones de tiempo solamente el 53% de las veces.

A pesar de esos problemas los resultados en el año 2000 no fueron aciagos. Por ejemplo, las exportaciones en el año 2000 en Brasil alcanzaron los 40 millones de dólares, en Chile los 15 millones y en la India sobrepasan los 7000 millones; otros datos como ventas, productividad, etc. en varios países aparecen en el Anexo 1. [CITMA, 2002] [Moreno, 2003] [García, 2000]

Países como Irlanda, India e Israel e inclusive algunos de América Latina, como México han podido posicionarse como jugadores de importancia en el negocio de la informática. Sin embargo, Colombia a pesar de algunos esfuerzos individuales y aislados no ha tenido el éxito esperado. Está ingresando un poco tarde a éste mercado, puesto que algunos países cuentan con corporaciones de miles de ingenieros y empleados que trabajan con éxito en la máquina de bits.

Estudios hechos estiman que para el año 2008 India exportará 57 billones de dólares en software y servicios asociados y que dicha industria dará empleo a 4 millones de personas.

Según los más importantes analistas extranjeros del mercado de informática y de acuerdo con las opiniones de los ejecutivos de las más importantes empresas de equipos y software, se avecina un enorme crecimiento para la contratación de software en el extranjero por parte de compañías de países industrializados. Algunas predicciones indican que para el año 2015 cerca de 3.3 millones de puestos de trabajo en el sector informático y contratos del orden de 136 billones de dólares se desplazarán de Estados Unidos a otros países que cuenten con compañías de calidad en el área de software.

El fomento y desarrollo de la industria del software está permitiendo a países menos desarrollados, posicionarse como proveedores de Tecnología de Información y Comunicaciones (TIC). Los resultados de algunos países con mayor éxito en esta industria se sintetizan más adelante en el siguiente epígrafe. [Oktaba, 2003]



### 1.1.1 La Industria de Software en Latinoamérica

Las empresas y los grandes compradores de software en Latinoamérica invierten desde hace algunos años grandes cantidades de dinero en la mejora de sus procesos de desarrollo, para poder unificar los criterios de la evaluación de la calidad y aspirar a la certificación cumpliendo las normas Organización Internacional para la Estandarización (ISO 9000) o el modelo estadounidense conocido como Modelo de Madurez de las Capacidades (CMM).

Existen iniciativas exitosas de apoyo al sector software en varios países de Latinoamérica:

Brasil, México, Chile, Perú, Costa Rica, Ecuador, Argentina

Tabla 1.1 Datos generales de la industria del software en algunos países de Latinoamérica [Takahashi, 2001] [Torres, 2001] [Meneses, 2001] [CaproSOFT, 2002]

País	Cantidad de empresas desarrolladoras de software	Evaluaciones y certificaciones (ISO, CMM y CMMI)	Instituciones rectoras
Brasil	3600 +	25 empresas certificadas ISO. 39 evaluaciones CMMI	ASSESPRO
México	206 +	Menos de 10 evaluaciones CMMI.	AMITI
Chile	342 +	Menos de 10 evaluaciones CMMI (4) y 21 evaluaciones CMM.	ACTI
Perú	150 +	No reporta evaluaciones CMMI	APESOFT
Costa Rica	120	No reporta evaluaciones CMMI, 2 evaluaciones CMM y 2 empresas de Software certificadas ISO 9000.	CAMTIC
Ecuador	223	No reporta evaluaciones CMM, ni CMMI.	AESOFT



		3 empresas están certificadas ISO 9000	
Argentina	1900	15 evaluaciones CMMI	CESSI

Los principales problemas a los que se enfrenta la industria de software en latinoamérica en estos momentos pueden clasificarse en tres categorías:

- Déficit de recursos humanos en cantidad y calidad.
- Fortalecimiento de una institución como representante de los intereses de los productores de software.
- La implementación de estándares de calidad de categoría internacional.

Dentro de estas se destacan dos variables que necesariamente deben considerarse para el desarrollo de un software: la calidad y la productividad. La primera se encuentra afectada por factores como el personal a cargo del proceso, el tipo de herramientas utilizadas y las metodologías aplicadas. El segundo aspecto se refiere a la capacidad de la empresa de reducir costos previniendo las fallas antes que se produzcan. [Baeza, 1995]

Entre las acciones que se están realizando en estos países latinoamericanos se encuentra la elaboración de una norma nacional que haga referencia a aspectos técnicos y administrativos promoviendo la realización de programas y certificados de calidad, así como la divulgación e implantación en la industria de estándares internacionales de calidad.

### 1.1.2 Las tres “I” en el desarrollo de Software

En los últimos años La India, Irlanda e Israel han tenido un desarrollo vertiginoso en la industria del software. Estos países proporcionan una gran variedad de servicios a la industria, inclusive realizan la parte más sofisticada del desarrollo del software. [Jalote, 2002a].



Tabla 1.2 Organizaciones de Israel, Irlanda y La India

País	Total de organizaciones registradas
India	430
Israel	1500-2000
Irlanda	840

Según P. Jalote, el éxito de la industria del software en la India ha sido el tránsito por la calidad, comenzando por productos, después proyectos, normas ISO y el modelo CMM. [Jalote, 2002a].

Las claves fundamentales que convirtieron a Israel en un actor importante en el desarrollo de software están asociadas con el alto desarrollo en la industria de capitales de riesgo, la alta disponibilidad de recursos capacitados y motivados, la necesidad de desarrollo de tecnología militar avanzada, las instituciones académicas con reconocimiento mundial y los programas activos por parte del gobierno, centros de investigación y desarrollo de incubadores tecnológicos.

En el caso de Irlanda estas claves fueron: la ubicación geográfica privilegiada para convertirse en proveedor de los mercados internacionales, la disminución de los costos del transporte, las tasas de interés a nivel internacional y la alta rentabilidad lograda. [Cadie, 2001].

## 1.2 Estado actual de la Industria Cubana de Software

Dentro de las proyecciones del gobierno cubano se encuentra el fomento de la InCuSoft, que permita diseñar y proveer de equipos electrónicos y sistemas informáticos que beneficien a la sociedad y también con posibilidad de exportarlos para aportar a la base material de todos los programas del país. [González, 2003]

En el año 2001, existían en el Ministerio de la Informática y las Comunicaciones (MIC) 24 empresas o entidades que tenían en su objeto social la producción de software y servicios informáticos, agrupadas en el Grupo de Tecnologías de Información (GTI). Esta estructura sufrió diversas reorganizaciones,



incluyendo la desaparición de este grupo. Actualmente hay 8 entidades en el MIC que desarrollan esta actividad. En este año se detectaron las siguientes dificultades:

Bajos niveles de explotación del software nacional en el país. El "negocio" para las empresas era representar productos y no existía un sistema que garantizara la calidad de estos con óptica de industria. Descalificación o pérdida del personal ("robo de cerebros"). Graduados con buena formación pero en número insuficiente. No se utilizaba correctamente la fuerza de técnicos medios. Poco desarrollo de productos para ambientes de software libre. Conceptos de soporte y mantenimiento atrasados sin explotar las posibilidades on-line, los call center, desk help y otros. Falta de integración real trabajando alrededor de un objetivo. Han primado los intereses individuales.

El mercado nacional y en frontera se abarcaba sin un ordenamiento, existiendo duplicidad de esfuerzos en muchos productos y servicios. La demanda nacional no se satisfacía, solo en el 2004-2005 es que se promulgan resoluciones ordenando que la venta de software y servicios informáticos a empresas estatales se realizaría en moneda nacional. Luego se produjeron exportaciones decrecientes hasta el año 2004, el pico se produjo en el 2001 con 8 MM USD y la presencia fundamental se logró con el software asociado a equipos médicos de alta tecnología.

En ese sentido, y observando las enormes ventajas que puede representar para un país el desarrollo de los servicios informáticos y la producción de software, el ex-ministro cubano de Informática y Comunicaciones, Ignacio González, ha expresado que Cuba debe aspirar a hacerse un espacio en el mercado mundial de software. Un negocio que mueve más de 440.000 millones de euros al año y que podría suponer una importante fuente de recursos para Cuba. En el informe presentado por el Ministerio de Relaciones Exteriores de Cuba en la Cumbre Mundial para la Sociedad de la Información en el 2004 plantea:

"La Industria Cubana del Software está llamada a convertirse en una significativa fuente de ingresos para el país, como resultado del correcto aprovechamiento de las ventajas del alto capital humano disponible. La promoción de la InCuSoft en el ámbito internacional ha tenido como línea estratégica aprovechar la enorme credibilidad que tiene Cuba en sectores tales como la salud, la educación y el deporte. El



continuar la producción sostenida de software de alta calidad en prestaciones, imagen y soporte, para satisfacer las necesidades nacionales en estos sectores, tendrá una positiva repercusión en el incremento de la exportación”.

Vale añadir que no solo significaría un incremento de las exportaciones, también contribuirá a la mejora de estos servicios y al ahorro, por concepto de sustitución de importaciones de software.

Algunos ejemplos de empresas enfocadas a lograr mejoras continuas de procesos de software lo constituyen la Empresa de Telecomunicaciones de Cuba (ETECSA) y el Centro de Calidad del Software (CALISOFT). La primera encaminada a comercializar productos de software de alto valor para la Gestión de los Operadores de Telecomunicaciones y Servicios de excelencia en la Industria de Software basados en estándares internacionales; y la segunda como un centro integrador de universidades, empresas, centros de investigación para la mejora de procesos. Que contribuye al desarrollo de la industria de software en el país incrementando su competitividad a través de la difusión, la mejora continua y el conocimiento de la tecnología de la información. Es además, un filtro de calidad para el software que se compre, para ser implantados en las organizaciones del país y pretende la inclusión de Cuba, con éxito, en el mercado internacional del software, constituyendo así un tercero confiable. [Ruiz, 2007].

En resumen, el país observa como una posibilidad alcanzable la incorporación en el mercado mundial de software, y se muestran serios pasos encaminados a cumplir ese objetivo: la creación del MIC, la creación de la UCI, la revitalización y potenciación de los tecnológicos de informática y la Universalización de la Carrera de Ingeniería Informática, son ejemplos ilustrativos.

En aras de lograr esta ambiciosa meta, todos los esfuerzos invertidos en la mejora de la calidad de los procesos y productos, elevando las capacidades de la naciente industria cubana son agradecidos, pues no seguir y desarrollar estos aspectos podría comprometer y dar al traste con lo alcanzado hasta el momento y los planes futuros.

La universidad cubana, por su parte, tiene varios compromisos con la sociedad: formar profesionales capaces es uno de ellos; como también lo es, el desarrollo de trabajos investigativos que contribuyan a la



mejora de la industria y la economía nacional. Por lo que puede afirmarse que está llamada a contribuir con su potencial científico al cumplimiento de los planes que en el campo del desarrollo de software, ha trazado la dirección del país.

Siguiendo este principio, desde inicios del 2000, y teniendo como objetivo el desarrollo de la InCuSoft, en el Centro de Referencia de Ingeniería de Software (CRIS) del Instituto Superior Politécnico “José Antonio Echeverría” (CUJAE), se vienen desarrollando un grupo de investigaciones relacionadas con la madurez de las organizaciones y la calidad en los procesos de desarrollo y en el producto final.

La InCuSoft se ha propuesto alcanzar un lugar relevante dentro del mercado mundial. Esta meta se vería seriamente comprometida si desde ya no se empieza a dedicar esfuerzo a la realización de actividades de GCS. Por supuesto, nuestra industria es incipiente y no se puede comparar con las empresas del primer mundo. En el año 2004 se presentó la tesis de doctorado [Febles, 2004] titulada “MConfig.PM, Modelo de Referencia para la GC en la pequeña y mediana empresa de software” con el objetivo, de crear un modelo para aplicar la GCS en las empresas cubanas teniendo en cuenta las características de las mismas.

### **1.3 Estado actual de la Gestión de Configuración en la Universidad**

La UCI se crea en el 2002 como idea del Comandante en Jefe Fidel Castro Ruz. En septiembre del mismo año comienza el primer curso con 2000 estudiantes, la cual tiene como misiones: formar profesionales, comprometidos con su Patria, altamente calificados en la rama de la informática y producir software y servicios informáticos, a partir de la vinculación estudio-trabajo como modelo de formación.

En el redimensionamiento de la InCuSoft la UCI funciona como un agente de cambio logrando:

- La concentración en un mismo lugar de miles de especialistas en constante actualización y ejercicios de sus conocimientos.
- Edad promedio.
- Tecnología instalada.
- Garantía de continuidad.



- Poder de convocatoria para vincular los mejores especialistas e instituciones del país en torno a los proyectos.
- **Curso 2002-2003:** No existían proyectos ni claridad de cómo iba a enfrentar la UCI el tema de la producción. Se estudiaban los Parques Tecnológicos. Se gana una licitación en México, pero el proyecto no resulta dada la inmadurez de la organización.
- **Curso 2003-2004:** Se buscan proyectos en todos los organismos del país. Gran dispersión de pequeños proyectos. La fuerza de estudiantes está en 1er y 2do año. No están claras las prioridades. Primeros intentos de organización metodológica y de una estructura organizativa para la producción. Experiencia de impacto en Venezuela.
- **Curso 2004-2005:** Se crea la Vicerrectoría Primera para unir la atención a los procesos de formación y producción. Se crean en la Infraestructura Productiva (IP) direcciones de software para la salud, software educativo, calidad, exportación de software, comunicación visual (Diseño). Los proyectos más importantes se organizan y dirigen desde la IP. Entidades productivas externas se ubican en la IP: Empresa de Soluciones Informáticas (Softel), PROCYON. Se definen conceptos para la producción de la UCI. Se crea Alternativas Bolivariana para la Exportación de Tecnologías (ALBET). Se comienzan grandes proyectos en el marco del Convenio de Colaboración Cuba-Venezuela.
- **Curso 2005-2006:** Se consolida la IP, se definen las facultades como las ejecutoras de los proyectos y la IP como la estructura que se responsabiliza con el proceso productivo. La UCI enfatiza su papel como coordinadora y desarrolladora de proyectos en el marco de la Alternativa Bolivariana para las Américas (ALBA). Propuestas de alianzas y negocios.

Una de las estrategias dentro de la producción en la universidad fue la creación de una IP como órgano metodológico, regulador, controlador, balancista, y que presta servicios generales a todas las facultades. Funcionan Consejos de Producción, de Calidad, de Tecnología. [Ruiz, 2007]

Dentro de la constitución de la IP, se encuentra la Dirección de Calidad, que cuenta con laboratorios de certificación, grupo de auditorías, revisiones y aseguramiento de la calidad. La relación que existe



mediante la Federación Estudiantil Universitaria (FEU), entre esta dirección y las distintas facultades de la universidad es lo que conforma el Proceso de Calidad de la UCI (Fig.1.2.).

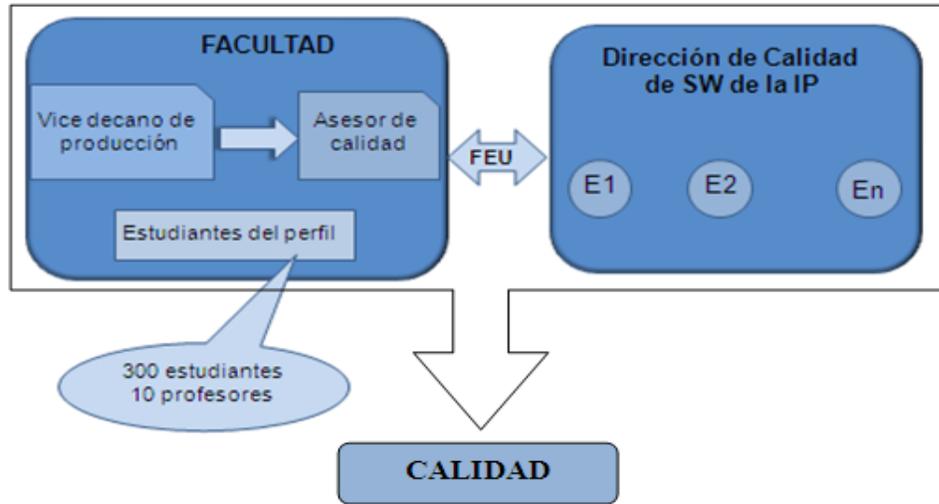


Figura 1.2: Proceso de Calidad en la UCI. Estructura

Dentro de las acciones estratégicas llevadas a cabo por la universidad se encuentra la creación de CALISOFT. Con el propósito de que contribuya al desarrollo de la InCuSoft y que permita la inclusión de Cuba en el mercado internacional del software. Además desarrolla un programa efectivo de mejora de procesos de software, basado en el modelo de calidad CMMI, certificando a especialistas con certificados internacionales y con la misión de crear Red Temática de Calidad y Mejora de Procesos.

La UCI es una universidad, y pretende ser una industria. Entonces, ¿cómo se produce con calidad y disciplina industrial en un entorno esencialmente docente? Esto se logra siguiendo los modelos de calidad internacionales que son de obligada referencia, pero hay que llegar a los modelos propios. El esquema de la Fig. 1.3 muestra como se está aplicando el modelo para evaluar el software en la UCI. [Ruiz, 2007]

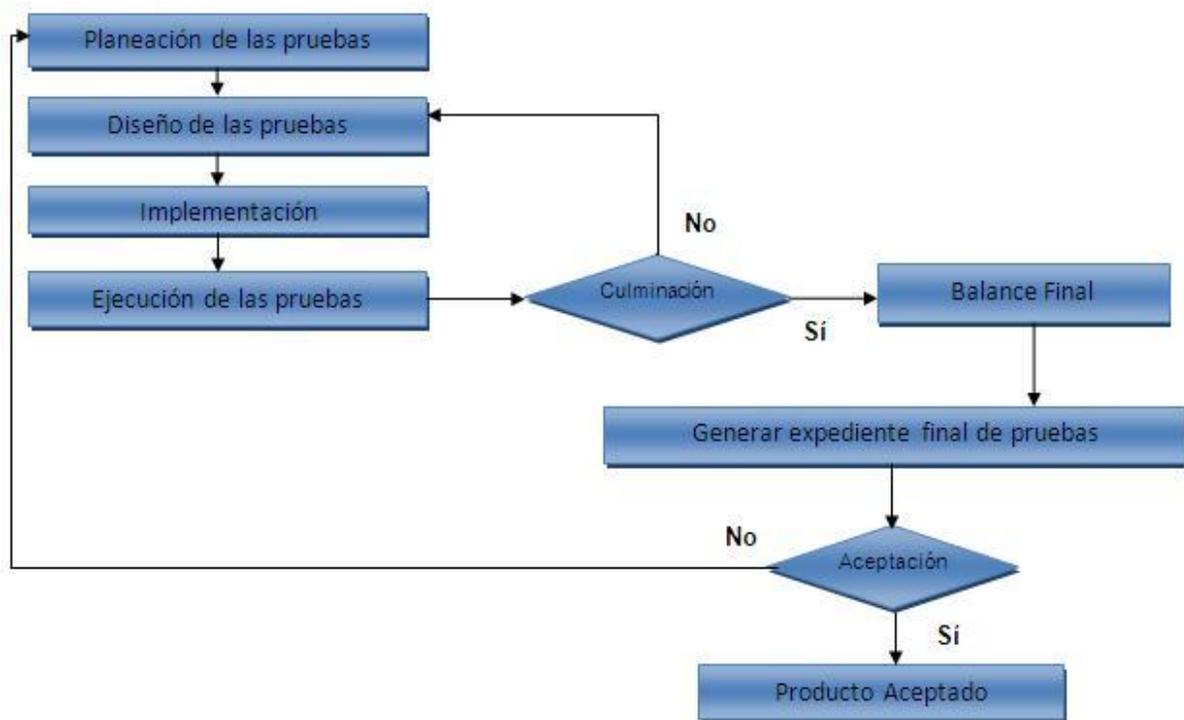


Figura 1.3 Plan de evaluación del Software

En la UCI, el tema de la GC es muy novedoso. Inicialmente en algunos proyectos productivos solo se realizaba el control de versiones usando algunas herramientas, principalmente el Subversion y el Visual Source Safe y el control de cambios a través del cliente, el cual no estaba definido y gestionado.

En los proyectos de la universidad, se utilizan herramientas para el control de versiones pero no se establecen procedimientos y políticas. Algunos usan herramientas, pero ellas solas no resuelven los problemas, tienen que ir acompañadas de políticas de uso, de conceptos establecidos. Los proyectos usan las herramientas pero no tienen implementados procedimientos por lo que no se lleva la GC como es debido.

En la actualidad dentro de las proyecciones de la dirección de calidad se encuentra el fomento y desarrollo de una mejor GC en cada proyecto. Esto se manifiesta en algunos como: Intranet de PDVSA, Sistema de



Gestión Penitenciaria (SIGEP), Centro Nacional de Tecnología de Información (CNTI), Cuerpo de Investigaciones Científicas, Penales y Criminalísticas (CICPC), los cuales han creado un equipo de aseguramiento de la calidad, con un personal dedicado solamente a la GC.

Los primeros pasos en este sentido han sido el control de cambio y la identificación de los elementos de configuración, utilizando herramientas como el Visual Source Safe y el Subversion; utilizadas principalmente por el proyecto Atención Primaria de la Salud (APS). Otros proyectos como Identidad y Registro y Notaria (R&N) realizan el control de versiones con el Subversion. [Delgado, 2007].

Muchos han sido los avances; pero faltan por especificar algunos aspectos, como la definición de las métricas y el control de release. Estas técnicas varían de acuerdo a las características de cada proyecto que permite medir el nivel de madurez alcanzado y el avance interno.

#### **1.4 Gestión de Configuración de Software. Definiciones**

Una definición en términos sencillos de GC es brindada por [Antonio, 2001]: "... disciplina, cuya misión es controlar la evolución de un sistema software" En realidad esta definición de GC es muy simple aunque bastante abarcadora.

Para Rational [Rational, 2003] la GC "describe la estructura del producto e identifica los elementos que lo constituyen y que son tratados como entidades que pueden ser puestas bajo control de versiones en el proceso de GC. La GC tiene que ver con la definición de la configuración, así como la construcción, el etiquetado y recolección de versiones de los artefactos".

Para Babich [Babich, 1986] "El arte de coordinar el desarrollo de software para minimizar la confusión se denomina GC. La GC es el arte de identificar, organizar y controlar las modificaciones que sufre el software que construye un equipo de programación. El objetivo es maximizar la productividad minimizando los errores."

Para otros autores la GC es la que administra y controla el contenido, el cambio o el estado de la información compartida en un proyecto.



[Brown 1998] El propósito fundamental de la GC es establecer y mantener la integridad y el control en los productos de software a lo largo del ciclo de vida del proyecto.

La definición de GC más utilizada es la que brindan el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) [IEEE, 1990] [IEEE, 1987] y que coincide con [Appleton, 2000] en la que establecen que:

“GC es la disciplina que abarca todo el ciclo de vida de la producción de software y productos asociados. Específicamente, requiere de la identificación de los componentes a controlar y la estructura del producto, controla todos los cambios sobre los elementos y garantiza mecanismos para auditar todas las acciones.”

Sin embargo, se considera que esta definición es aún incompleta [Febles 2004]. Pues faltan elementos como el control del proceso y el control del esfuerzo de los desarrolladores [Appleton, 2000].

Otra definición es propuesta por la Norma ISO 9000-3:1991 [Bamford, 1995], donde se establece que la GCS provee mecanismo para identificar, controlar y dar seguimiento a cada una de las versiones de los elementos que conforman al producto software.

De todas las definiciones revisadas para esta investigación, se ha considerado la más completa y a su vez concisa la de Ivar Jacobson, Martín Griss y Patrick Jonson en el libro **“Software Reuse: Architecture, Process, and Organizations for Business Success”** en la que plantean [Jacobson, 2000]:

“Gestión de Configuración: Proceso de soporte cuyo propósito es identificar, definir y almacenar en una línea base los elementos de software, controla los cambios, reporta y registra el estado de los elementos y de las solicitudes de cambio; asegura la completitud, consistencia y corrección de los elementos; controla, almacena, maneja y libera los elementos asociados al producto de software”.

### **1.5 Gestión de Configuración de Software. Valoraciones.**

A continuación se brindan apreciaciones de diversos autores que permiten apreciar cuanto se valora la GC, y qué nivel de importancia se le otorga dentro del desarrollo de software.



En [Racional, 2004], se hace referencia a una frase escrita por Tom Milligan asesor de International Business Machines (IBM) Rational Software, donde plantea que: “La GCS es el héroe no nombrado en el proceso de desarrollo de software”.

Watts Humphrey [Humphrey, 1989], en el libro, “Administrando el Proceso de Software” plantea que en el desarrollo de software, los problemas más frustrantes, son causa de un pobre proceso de GC. Y resultan frustrantes porque toman tiempo para solucionarlos, ocurren generalmente en momentos críticos y su ocurrencia es totalmente innecesaria.

Pressman designa en [Pressman, 1997] a la GC con el término en inglés “*umbrella activity*”, al ahondar en el uso del adjetivo “umbrella” encontramos que es empleado para actividades abarcadoras, globales, incluyentes, rectoras, notándose así el alcance que le atribuye el citado autor.

Para concluir no debe dejarse de mencionar la Ley Fundamental de la GC [cmBook, 1998], donde se define el papel de la GCS en el proceso de desarrollo:

“La GC es el fundamento de un proyecto software. Sin ella, no importa cuan talentoso sea el equipo, cuan grande sea el presupuesto, cuan robusto sean los procesos de desarrollo y prueba, o cuan superior sean las herramientas de desarrollo técnicamente, la disciplina del proyecto colapsará y se perderá la posibilidad de triunfo. Haz bien la GC, y olvídate de avanzar en el proceso de desarrollo de Software”.

## **1.6 Modelos y estándares tenidos en cuenta en la investigación**

### **1.6.1 Gestión de Configuración del Software en los Modelos de Calidad.**

Según Pressman [Pressman, 1998] la calidad de software es “la concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados y con las características implícitas que se esperan de todo software desarrollado profesionalmente”.



Para la Gestión de Calidad de Software a nivel mundial se han seguido principalmente dos tendencias: la primera a seguir las reglas implantadas por las oficinas internacionales de estandarización para los productos y servicios a través de las normas ISO y la IEEE, y la segunda a seguir las creadas específicamente para el mundo del software como CMMI y Determinación de la mejora de Proceso y de la Capacidad del Software (SPICE).

#### **1.6.1.1 IEEE (Instituto de Ingenieros Eléctricos y Electrónicos)**

A la hora de hablar de los estándares para la calidad de software es prácticamente imposible dejar de mencionar el estándar IEEE 730-1998 para el Plan de Aseguramiento de la Calidad de Software [IEEE, 1998]. Este aborda aspectos como la administración, la documentación, el control de código, etc. Dentro de la documentación le adjudica gran importancia al Plan para la GCS reflejado en el estándar IEEE 828-1998 [IEEE, 1998].

Dicho plan trata todo lo referente a la asignación de las responsabilidades, la identificación de las actividades que se realizarán durante todo el proceso, la identificación de la configuración, el reconocimiento de los elementos de configuración, el control de la configuración, el acceso a las bibliotecas, la aprobación o desaprobación de un cambio y la implementación del cambio de ser aprobado (todas estas incluidas entre las tareas de la GC).

#### **1.6.1.2 CMMI Integración del Modelo de Madurez de las Capacidades.**

Al inicio de la década de los 90, surgieron una serie de iniciativas para aplicar las mejores prácticas en el ámbito del desarrollo del software que se plasmaron en la realización y publicación de una serie de modelos y normas para el sector. Uno de estos modelos fue CMM desarrollado por el Instituto de Ingeniería de Software (SEI) en el año 1991. CMM para software o SW-CMM no es más que un grupo de lineamientos para el desarrollo de software [Gartner, 2001]. Una descripción de los escenarios por los cuales las organizaciones de software evolucionan, cómo ellos definen, implementan, miden, controlan y desarrollan sus procesos de software.



Después del lanzamiento del modelo SW-CMM se desarrollaron otros modelos para otras disciplinas. Pronto las empresas se encontrarían que resultaba muy difícil integrar todos estos modelos, solucionar las lagunas detectadas, resolver las inconsistencias y aclarar las diferentes terminologías.

Para dar solución a estos problemas surge en diciembre del 2001 CMMI. [CMMI, 2005] el cual es un enfoque para la mejora de procesos que brinda a las organizaciones los elementos básicos sobre procesos efectivos. Puede ser empleado para guiar la mejora de procesos a lo largo de un proyecto, división u organización. CMMI ayuda a integrar funciones organizacionales tradicionalmente separadas, otorga metas y prioridades en la mejora de procesos, brinda además una guía para los procesos específicos de calidad y un punto de referencia para la valoración de los procesos actuales [CMMI, 2005]. Cuenta al igual que CMM con cinco niveles, aunque algunos cambian de nombre [Ema, 2002].

Para poder escalar a un nivel superior es necesario haber cumplido todos los aspectos que plantea el nivel actual, por lo que es posible incluso, cumplir con aspectos de nivel 2 y de nivel 3 simultáneamente y, aún así, mantenerse en un nivel 1. Para obtener Nivel 2 de CMMI (Gestionado) es necesario realizar las siguientes tareas [Ema, 2002]:

- Gestión de los requisitos del producto y del proyecto
- Planificación de los proyectos
- Seguimiento y control de los proyectos
- Gestión de acuerdos con los proveedores de productos y servicios
- Selección y supervisión de los proveedores
- Medición y análisis
- Aseguramiento de la calidad del producto y del proceso
- Gestión de Configuración de Software.

Cada una de estas áreas se divide en varios objetivos y prácticas que hay que cumplir para llegar al nivel de madurez indicado.



### **Gestión de requisitos**

- Obtener y gestionar los requisitos
- Obtener la aprobación de los requisitos
- Gestionar los cambios de los requisitos
- Mantener la trazabilidad bidireccional de los requisitos
- Identificar las inconsistencias entre el trabajo real a realizar y los requisitos

### **Planificación de proyectos**

- Establecer estimaciones
- Desarrollar un plan de proyecto
- Obtener el compromiso con el plan

### **Control y seguimiento de proyectos**

- Control del proyecto contra el plan: informes de estado y reuniones periódicas
- Gestionar las acciones correctivas

### **Gestión de acuerdos con proveedores**

- Establecer los acuerdos con el proveedor
- Satisfacer los acuerdos con el proveedor

### **Medición y análisis**

- Establecer las medidas y actividades de análisis
- Proporcionar los resultados de la medición y analizarlos

### **Aseguramiento de la calidad**



- Evaluar los procesos y productos de trabajo objetivamente
- Proporcionar los resultados obtenidos y proponer acciones correctivas

### **Gestión de configuración**

- Establecer las líneas base
- Rastrear y controlar los cambios
- Establecer la integridad de las líneas base

Vuelve a ponerse de manifiesto como tiene un lugar básico la GC, pues de los cinco niveles que plantea el modelo, desde el número 2 hay que garantizar las actividades de GC para poder considerar que se cumple con la especificación del mismo.

#### **1.6.1.3 ISO. Organización Internacional de Normalización.**

ISO es una red de institutos nacionales de estándares de 156 países que promueve la normalización internacional para facilitar el intercambio de bienes y servicios como de aplicaciones.

Relacionado con los procesos de software ha sido implementado el modelo de evaluación y mejora del proceso de software ISO 9000, específicamente la guía ISO 9000-3.

Entre los aspectos que tiene en cuenta la Norma ISO 9000-3 se encuentran:

- Sistema de calidad
- Especificación de los requisitos del comprador
- Planificación del desarrollo
- Planificación de la calidad
- Pruebas y validaciones
- Gestión de Configuración de Software
- Control de documentos
- Mediciones



El hecho de que una norma de calidad tan reconocida a nivel mundial incluya dentro de sus requerimientos, el de efectuar las labores de GCS, es plenamente consistente con la importancia que le otorgan los diversos autores analizados, lo que puede tenerse una conclusión parcial hasta este punto. Aunque aún no han sido expuestas todas las actividades concretas que establece la GC, a partir del análisis bibliográfico efectuado, se puede valorar la importancia dentro del proceso de desarrollo por autores de gran prestigio, y consecuentemente es considerada en las principales normas y modelos de calidad de software a nivel mundial.

### **1.6.2 Investigación sobre los modelos existentes a nivel mundial**

Varios autores afirman que los procesos asociados a la GCS que aparecen en el estándar de la IEEE son suficientes para conseguir sus objetivos principales, [Gervás, 2002] [Antonio, 2001] [IEEE, 1990] [IEEE, 1987] estas actividades son las siguientes:

- Identificación de la Configuración
- Control de Cambios en la Configuración
- Generación de Informes de Estado
- Auditoria de la Configuración

Con igual propósito en CMM, para la GC, se proponen los procesos o prácticas claves siguientes [Paulk, 1993] [Paulk, 1999a]:

- Planificación de las actividades de GC
- Identificación de los ECS
- Control de cambios a los ECS
- Informar a los grupos e individuos involucrados de los cambios a los ECS
- Auditoria de la Configuración

Por otra parte ISO sugiere para esta área los siguientes procesos [ISO, 1995]:



- Identificación de la configuración
- Control de cambios a la configuración
- Informe del estado de la Configuración
- Auditoria de la configuración

Sin embargo esta investigación considera que las tres propuestas, individualmente son incompletas. En el Capítulo 3 se proponen y fundamentan los procesos que a partir de la investigación realizada son incluidos en el modelo.

La ausencia de una GC en los procesos de desarrollo de software, deriva en la mala utilización del recurso humano altamente calificado, provocando atrasos en actividades de desarrollo y mantenimiento de los productos, y un inadecuado y a veces inexistente control de proyectos. Por lo que se considera imprescindible para el desarrollo de la InCuSoft establecer un sistema de GC que sea implantado y ejecutado en la UCI.

### **1.6.3 La vinculación de estándares**

En general, existe una estrecha relación de todos estos modelos aunque difieren en elementos particulares. En particular en el caso de CMM e ISO, las organizaciones que se correspondan con el nivel 1 de CMM pueden cumplir con las exigencias de las certificaciones de ISO. Esto ocurre, porque las organizaciones pudieran cumplir con la mayoría de las exigencias del nivel 2 pero no con todas y con algunas del nivel tres. Esto implica que no se puede decir que estén justamente en ninguno de estos niveles, sin embargo, los aspectos con los que cumplen en ambos niveles podrían resultar suficientes para certificarse como ISO (Fig. 1.4). Entonces pudiera haber organizaciones no maduras (CMM) o por debajo del nivel 3 que obtuvieran la certificación ISO.

De manera general, si una organización se considera posible candidata a ISO, es probable que esté cerca de alcanzar el nivel 2. Las que están en el nivel 3 con completa seguridad serán certificadas ISO9001. [Zhang, 2001]

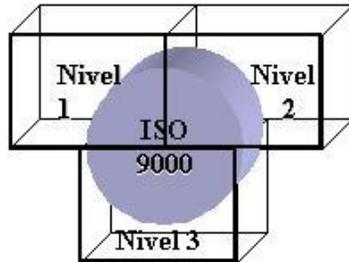


Figura 1.4 Vinculación de ISO y CMM

#### 1.6.4 Un proceso para el desarrollo de software (RUP)

Rational Unified Process (RUP), es un proceso de IS propuesto por Rational Software Corporation para la construcción completa del ciclo de IS. Permite la productividad en equipo y la realización de mejores prácticas de software a través de plantillas y herramientas que lo guían en todas las actividades de desarrollo del software.

Es un producto que unifica las disciplinas en lo que a desarrollo de software se refiere, incluyendo modelado de negocio, manejo de requerimientos, componentes de desarrollo, ingeniería de datos, manejo y configuración de cambios, y pruebas, cubriendo todo el ciclo de vida de los proyectos basado en la construcción de componentes y maximizando el uso del UML (**Unified Modeling Language**).

Pueden resumirse tres aspectos definatorios de RUP: está dirigido por casos de uso, centrado en la arquitectura, e iterativo e incremental. Lo que hace único al proceso. Para hacer que estas ideas funcionen, se necesita un proceso polifacético, que tenga en cuenta ciclos, fases, flujos de trabajo, gestión de riesgo, control de calidad, gestión de proyectos y GC. [Jacobson, 1997].



## **Conclusiones parciales**

- La industria del software es una industria creciente a nivel mundial, no obstante aún existen problemas relacionados con la calidad del producto final, la definición y ejecución de procesos y la satisfacción del cliente.
- En la literatura consultada varios autores consideran la GC como el proceso central en la producción de software que debe estar presente en todo el ciclo de vida. Sin embargo, según estadísticas internacionales y encuestas nacionales; es un proceso no conocido, que no se define en las organizaciones y por lo tanto tampoco se ejecuta correctamente, trayendo como consecuencia pérdidas de información, gastos innecesarios de recursos, insatisfacción del cliente y desconocimiento de los directivos y líderes sobre el estado real de los proyectos.
- Los modelos de calidad internacionales son de obligada referencia, pero hay que llegar a la elaboración de modelos propios.

Por todo ello es que constituye una necesidad, para la InCuSoft y en particular para la UCI, definir un modelo de referencia para la GC, el cual incluirá un conjunto de procesos, métricas y herramientas automatizadas que permitirán de forma eficiente implantar la GC en Cuba. El modelo sienta las bases para poder tomar acciones correctivas, garantizar mejor comunicación entre los involucrados, calidad en la ejecución de los procesos y mayor calidad en el producto final.



## **Capítulo II: La Gestión de Configuración en la UCI.**

La UCI se crea en el 2002, con la misión de formar profesionales comprometidos con su Patria, altamente calificados en la rama de la informática, producir software y servicios informáticos, a partir de la vinculación estudio-trabajo como modelo de formación, diseñada para lograr resultados concretos. Se diseña la vinculación del estudiante a los proyectos desde el 1er año, en roles más sencillos, hasta el quinto curso ya en roles de alta complejidad.

Los proyectos productivos de la universidad están destinados a la informatización del país, la informatización de la vida de la propia universidad, y la exportación. Los cuales se organizan, gestionan, ejecutan y dirigen en las facultades. Estas asumen los proyectos según las áreas en que se hacen expertas. En la UCI se crea una IP como órgano metodológico, regulador, controlador, balancista, y que presta servicios generales a todas las facultades. Funcionan Consejos de Producción, de Calidad, de Tecnología.

Debido al gran compromiso que se tiene de desarrollar la economía del país a través de la producción de software es que se necesita que tengan la mayor calidad posible. Como se explicó anteriormente existe una gran variedad productiva por lo que se urge crear un modelo estándar y eficiente para medir la calidad de todos los productos. Para lograr la calidad máxima se decide hacer un modelo de GC aplicable a todos los productos.

Con este objetivo se aplicó una encuesta en todas las facultades a una selección de algunos proyectos productivos de ella; se propuso como un instrumento para obtener información cuantificable y con validez estadística sobre la aplicación de la disciplina GC, y arribar a conclusiones que permita la confección de un modelo aplicable a todos los proyectos independientemente de las características que este presente. El listado de los proyectos productivos seleccionados aparece en el Anexo 2

Los resultados se presentarán en el desarrollo de este capítulo.



## **2.1 Técnicas empleadas**

### **2.1.1 La entrevista**

Es un método de recopilación de datos empíricos, un proceso de comunicación entre dos o más personas, generalmente de forma oral donde las preguntas al entrevistado se hacen por vía directa, se deben desarrollar preguntas que permitan respuestas precisas. [Guido, 2005]

### **2.1.2 La Revisión de bibliografía**

Se utiliza para recoger la información que se encuentra registrada en un documento establecido. Se utilizaron los listados de estudiantes por proyectos, listado de la relación de proyectos por facultades, etc. [Guido, 2005]

### **2.1.3 La encuesta**

Es un diseño o estrategia general no experimental, que permite contrastar las hipótesis de investigación con información sobre características de una población determinada. En efecto, constituye una de las formas más utilizadas para indagar sobre las características principales de un segmento poblacional y/o para conocer las opiniones, experiencias y expectativas de grupos de personas. Esta técnica permite recabar una serie de datos en un amplio número de casos. [Guido, 2005]

## **2.2 Etapas en la elaboración de una encuesta.**

La implementación de encuestas supone el seguimiento de un procedimiento que se inicia con el diseño (formulación de los objetivos e interrogantes, elección de opciones metodológicas) y continúa con la construcción de muestras, en el caso en que se decida guardar representatividad estadística. De forma paralela se operacionalizan las dimensiones y variables bajo indagación, a partir de su traducción en un cuestionario o guía de preguntas. Por último, el procedimiento de implementación de encuestas conduce a la puesta en práctica de la recolección de datos por diversas vías (depende del tipo de metodología escogida), el procesamiento, el análisis de la información y la presentación final de los resultados, como se muestra en la Tabla 2.1. [Guido, 2005]



Tabla 2.1 Etapas para el diseño y la elaboración de una encuesta

Etapas para el diseño y elaboración de una encuesta	
I.	Formulación de objetivos
II.	Diseño de la muestra
III.	Confecciones del cuestionario
IV.	Trabajo de campo (Recolección de datos)
V.	Tareas posteriores a la recolección de datos
VI.	Procesamiento, análisis de los datos y presentación de los resultados

### 2.2.1 Etapa I: Formulación de objetivos.

1. Definir los propósitos del estudio o la investigación y los objetivos de la encuesta.

El presente estudio se hizo con la finalidad de conocer el comportamiento en la UCI de la disciplina de GCS. De acuerdo a los resultados que se obtuvieron es que se pretende crear un Modelo de GCS con el propósito de guiar a los proyectos productivos hacia la búsqueda del éxito en la calidad. Se ha convertido en un punto de interés en la administración de las diferentes compañías del mundo. Constituye un elemento de importancia en la decisión del cliente para la selección de un producto o servicio y una vía para alcanzar el éxito organizacional y el crecimiento hacia mercados nacionales e internacionales.

Con el objetivo de darle cumplimiento a lo anteriormente expuesto es que se necesita conocer los siguientes tópicos:

- I. ¿Qué conocimiento tienen los estudiantes y líderes de proyectos de la GCS?
- II. ¿Cómo se aplica la disciplina de GCS en los diferentes proyectos productivos de las facultades?
  - Si tienen un personal dedicado específicamente a la GCS.
  - Si tienen en su proyecto el rol de Administrador de Configuración.



- III. Cómo se identifican y gestionan las versiones existentes de los programas y documentación en los proyectos y como se controlan los cambios antes y después de que el software sea distribuido al cliente. En ambos casos se debe especificar las herramientas utilizadas.

### 2.2.2 Etapa II: Diseño de la muestra.

Una vez definido los objetivos del estudio y la población sobre la cual se aplicará la encuesta, se definió la estrategia de aplicación. Se decidió seleccionar una muestra de la población, se tuvo en cuenta el nivel de madurez de los proyectos productivos. Se escogieron proyectos representativos de mayor experiencia y productividad, los que no son tan productivos y los que estaban surgiendo en el momento del procesamiento.

Se realizó un Muestreo Probabilístico, y dentro de este fue seleccionado el Muestreo Aleatorio Estratificado (MAE). Consiste en subdividir la población en sub-poblaciones de tal forma que la unión de ellos será la población, y la intersección de cualquiera de los dos dará como resultado el conjunto vacío, es decir, no tendrán elementos comunes.

A las sub-poblaciones se les llamará estratos, se tratará de conformar estos de modo que los elementos dentro de ellas sean homogéneos. El tamaño de la muestra se distribuirá entre los estratos, en función de distintos criterios pero lo que caracteriza al MAE es que la selección de la muestra de cada estrato se hará bajo el procedimiento de muestreo irrestricto aleatorio y se realizará independientemente de los diferentes estratos. Es recomendable estratificar en función del tamaño de las unidades y distribuir la muestra proporcionalmente al número de unidades de los estratos. [Guido, 2005]

#### 2.2.2.1 Notación

Se supone que la población consta de  $n$  unidades y están distribuidas en  $L$  estratos, constituyen una partición de la población; se representará por  $N_h$  en el  $n_i$  de  $u$  en el estado  $h$ -ésimo, de aquí:

$$N = N_1 + N_2 + \dots + N_h + \dots + N_l$$



Total de la población  $x_h = \sum_1^{N_h} x_{hi}$  media  $\bar{x}_h = \frac{1}{N_h} \sum_1^{N_h} x_{hi}$

$f_h = \frac{n_h}{N_h}$  Fracción de muestreo del estrato.

Si el tamaño de la muestra de los estratos se distribuye proporcionalmente al número de unidades en el estrato, es decir se cumple que:

$n_h = n \frac{N_h}{N}$  para todo h. En este caso se dice que la distribución de la muestra se ha hecho con asignación proporcional.

Para la determinación del tamaño de muestra de una población con  $S^2$  desconocida se puede determinar mediante la expresión:

$$n = \frac{\left( \frac{Z_{1-\frac{\alpha}{2}}}{d} \right)^2 P \llcorner - P \rceil}{1 + \frac{1}{N} \left( \frac{Z_{1-\frac{\alpha}{2}}}{d} \right)^2 P \llcorner - P \rceil - \frac{1}{N}}$$

Donde  $1-\alpha$  es el nivel de confianza,  $d$  es el error

absoluto, Z percentil de la distribución normal, P proporción de la población y N tamaño de la muestra.



Realizando el cálculo del tamaño de muestra para un nivel de confianza del 90%, con un error absoluto de 0.10, se obtiene el percentil de la distribución normal de 1.64 y se asume como proporción de la población  $P = 0.5$ , de donde se obtiene: [Toirac, 2007].

$$n = \frac{\left(\frac{1.64}{0.1}\right)^2 \cdot 0.5 \cdot (1 - 0.5)}{1 + \frac{1}{1170} \cdot \left(\frac{1.64}{0.1}\right)^2 \cdot 0.5 \cdot (1 - 0.5) - \frac{1}{1170}}$$

Calculando el tamaño de muestra por estrato (se considera estrato a cada región) aplicando  $n_h = n \frac{n_h}{N}$  resultan los tamaños de muestra por estrato que se muestran en la Tabla 2.2.

Los grupos de estudiantes por proyectos productivos son los siguientes:

Tabla 2.2 Tamaños de muestra por estratos de la población.

Estratos	Cantidad	Tamaño de la Muestra
Akademos (Facultad 1)	40	2
ECC – IPOSTEL (Facultad 1)	62	3
Identidad (Facultad 1)	70	3
PROCYON (Facultad 2)	24	1
Operación Verdad (Facultad 2)	34	1
CTAISC (Facultad 2)	59	3
Inventario (Facultad 3)	33	1



ONE (Facultad 3)	65	3
Mixta Cuba Venezuela (Facultad 3)	25	1
SIGEP (Facultad 4)	67	3
MINFAR (Facultad 4)	109	5
Quirúrgico (Facultad 5)	44	2
SCADA de PDVSA (Facultad 5)	65	3
Ensayos Clínicos (Facultad 6)	67	3
(GRAPh-TOol) (Facultad 6)	36	1
(BioSyS) (Facultad 6)	25	1
GPI (Facultad 7)	38	2
APS (Facultad 7)	73	3
CNTI (Facultad 8)	15	1
CICPC (Facultad 8)	105	5
A Jugar (Facultad 9)	13	1
Operación Verdad (Facultad 9)	24	1
Control y Gestión de Indicadores (Facultad 9)	20	1
Nova (Facultad 10)	21	1
Archivo General de la República de Venezuela (Facultad 10)	26	1
Intranet de PDVSA (Facultad 10)	10	1



### 2.2.3 Etapa III: Confección del cuestionario.

Esta etapa implica la selección de las dimensiones o variables que permitirán operacionalizar los objetivos y elaborar el instrumento de recolección de datos (cuestionario). Este combina preguntas cerradas y abiertas. Las preguntas son expuestas claramente y sin ambigüedades, para evitar malas interpretaciones, las mismas se presentan en el Anexo 3. [Guido, 2005]

#### 2.2.3.1 Clasificación de variables

Una variable es una función que asocia a cada elemento de la población la medición de una característica, particularmente de la característica que se desea observar.

De acuerdo a la característica que se desea estudiar, a los valores que toma la variable se dan varias clasificaciones referenciadas en la Fig. 2.1.

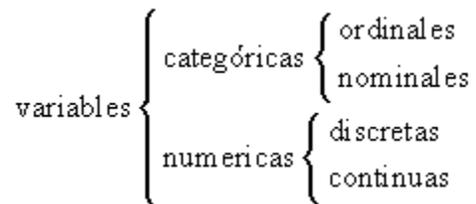


Figura 2.1 Clasificación de variables

En este caso en específico las variables que se utilizaron fueron:

1. Las variables **categóricas** son aquellas cuyos valores son del tipo categórico, es decir, que indican categorías o son etiquetas alfanuméricas o "nombres". A su vez se clasifican en:

- a) Variables categóricas nominales: son las variables categóricas que, además de que sus posibles valores son mutuamente excluyentes entre sí, no tienen alguna forma "natural" de ordenación. Por ejemplo, cuando sus posibles valores son: "sí" y "no". A este tipo de variable le corresponde las escalas de medición nominal.



2. Las variables **numéricas** toman valores numéricos. A estas variables le corresponde las escalas de medición de intervalo, y a su vez se clasifican en:

- a) Variables numéricas discretas: son las variables que únicamente toman valores enteros o numéricamente fijos. Por ejemplo: las ocasiones en que ocurre un suceso, la cantidad de pesos que se gastan en una semana, los barriles de petróleo producidos por un determinado país, los puntos con que cierra diariamente una bolsa de valores, etcétera. [Guido, 2005]

### 2.2.3.2 Estructura de un Cuestionario Tipo

En este sub epígrafe se expone la temática general que está presente en el cuestionario elaborado para proyectos productivos de la UCI.

### 2.2.4 Etapa IV: Trabajo de campo (recolección de los datos)

Se denomina trabajo de campo al conjunto de las acciones que tienden a localizar a las personas que deben contestar las preguntas, la gestión y la administración de los cuestionarios o métodos alternativos de recolección de la información deseada.

De las principales metodologías que existen para la recolección de los datos a través de encuestas se usaron:

a) **Encuestas personales:** implican formular preguntas directas cara a cara a un grupo de personas. Este tipo de encuestas otorga mayor flexibilidad en la obtención de la información, ya que permite gran libertad en cuanto al formato y la longitud del instrumento. Por otro lado, la interacción directa facilita la comprensión de las preguntas, permite aclararlas e introducir variantes (mostrar objetos o fotografías) que otro tipo de encuestas no permite.

b) **Encuestas auto-administradas:** esta metodología supone el llenado del cuestionario por parte de los propios ciudadanos sin asistencia de encuestadores. Por tal motivo, requiere un diseño claro y fácilmente comprensible. Generalmente se usan cuando no se trabaja con muestras estadísticamente



representativas, ya que muchas veces no puede controlarse quiénes son los que responden efectivamente. [Guido, 2005].

Cada metodología presenta una serie de ventajas comparativas que pueden resumirse en la Tabla 2.3\*.

Tabla 2.3 Ventajas comparativas para cada metodología.

	Encuesta Auto-administrada	Encuesta personal
Cantidad de datos que pueden recopilarse	Regular	Excelente
Flexibilidad	Deficiente	Excelente
Sesgo de la muestra (tasa de respuesta)	Deficiente a Buena	Excelente

\* *Elaborado en base a la Guía de Orientación para la Realización de Estudios de Análisis de la Demanda y de Encuestas de Satisfacción, Ministerio de Administraciones Públicas, Madrid, 2006.*

### 2.2.5 Etapa V: Tareas posteriores a la recolección de datos.

Una vez realizada la encuesta, se dio inicio a la depuración y grabación de los cuestionarios. Las primeras dos tareas se encuentran relacionadas con la necesidad de revisar la consistencia de los datos recolectados, evitando así errores u omisiones en el llenado de los cuestionarios. Una vez que estos trabajos se completaron, se inicia la grabación de los datos en la computadora. [Guido, 2005].

### 2.2.6 Etapa VI: Procesamiento, análisis y presentación de los datos.

Después de haberse cargado la totalidad de los datos en la computadora comienzan las tareas de procesamiento estadístico de los datos relevados. El procesamiento concluye una vez que se han obtenido las tablas o cuadros con los porcentajes de respuestas obtenidas en cada categoría.



En relación a la presentación de la información es necesario recordar, para organizar adecuadamente la misma, que las encuestas tienen por objetivo principal ayudar a los líderes y estudiantes de proyectos, en la toma de decisiones. Por lo tanto, la información final ha de ser muy clara incluyendo únicamente aquella relevante para las decisiones que hay que tomar. Se recomienda que una vez procesados los datos y obtenidos los cuadros correspondientes, el informe sea redactado con el apoyo de gráficos y cuadros, debidamente seleccionados para no abrumar de información al lector. En tal sentido, en ocasiones es preferible presentar un gráfico, dado que éste permite sintetizar la información contenida en más de un cuadro, facilitando así la lectura. [Guido, 2005]

Tabla 2.4: Por ciento de los estudiantes por proyectos productivos.

<b>Estratos</b>	<b>Tamaño de la Muestra</b>	<b>% de estudiantes</b>
Akados (Facultad 1)	2	5
ECC – IPOSTEL (Facultad 1)	3	4.83
Identidad (Facultad 1)	3	4.22
PROCYON (Facultad 2)	1	4.17
Operación Verdad (Facultad 2)	1	2.94
CTAISC (Facultad 2)	3	5.08
Inventario (Facultad 3)	1	3.03
ONE (Facultad 3)	3	4.61
Mixta Cuba Venezuela (Facultad 3)	1	4
SIGEP (Facultad 4)	3	4.45
MINFAR (Facultad 4)	5	4.71
Quirúrgico (Facultad 5)	2	4.54
SCADA de PDVSA	3	4.61



(Facultad 5)		
Ensayos Clínicos (Facultad 6)	3	4.47
(GRaph-TOol) (Facultad 6)	1	2.77
(BioSyS) (Facultad 6)	1	4
GPI (Facultad 7)	2	5.26
APS (Facultad 7)	3	4.10
CNTI (Facultad 8)	1	6.66
CICPC (Facultad 8)	5	4.76
A Jugar (Facultad 9)	1	7.69
Operación Verdad (Facultad 9)	1	4.16
Control y Gestión de Indicadores (Facultad 9)	1	5
Nova (Facultad 10)	1	4.76
Archivo General de la República de Venezuela (Facultad 10)	1	3.84
Intranet de PDVSA (Facultad 10)	1	10

A estos grupos de estudiantes se les aplicó una encuesta, Anexo 3 en la que se recoge diferentes opiniones acerca de cómo se lleva a cabo la disciplina de GC en los proyectos productivos de la UCI, con vista a detectar alguna irregularidad o el desconocimiento sobre el tema. Se decidió segmentarlos por facultades para evitar que sus diferentes perfiles puedan influir significativamente en los resultados, estratificándose entonces por los distintos proyectos.

A continuación se muestran las tablas con las preguntas y respuestas relacionadas con el cuestionario.



Tabla 2.5 ¿Se dedican solo a la producción de software?

	Sí	No
	18	24
%	42.8	57.1

Tabla 2.6 Otras actividades que se realizan en los proyectos productivos

Actividades	Total	%
I + D	8	19.0
Docencia	4	9.5
Mantenimiento y soporte	3	7.1
Multimedias	3	7.1
Web	2	4.7
Cursos de capacitación	2	4.7
Investigación	7	16.6
Calidad y planificación	1	2.3
Divulgación	2	4.7

Tabla 2.7 Cantidad de productos fabricados

Nombre de los Proyectos	Cantidad de productos fabricados	No sabe	%
por Facultades			
Akademios (Facultad 1)	1		2.3
ECC – IPOSTEL (Facultad 1)	0		0



Identidad (Facultad 1)		X	0
PROCYON (Facultad 2)	+10		23.8
Operación Verdad (Facultad 2)	0		0
CTAISC (Facultad 2)	0		0
Inventario(Facultad 3)	0		0
ONE(Facultad 3)	1		2.3
Mixta Cuba Venezuela(Facultad 3)	0		0
SIGEP (Facultad 4)		X	0
MINFAR (Facultad 4)	0		0
Quirúrgico (Facultad 5)	0		0
SCADA (Facultad 5)	0		0
Ensayos Clínicos (Facultad 6)	0		0
(GRaph-TOol) (Facultad 6)	0		0
(BioSyS) (Facultad 6)	0		0
GPI (Facultad 7)	12		28.5
APS (Facultad 7)	12		28.5
CNTI (Facultad 8)	2		4.7
CICPC (Facultad 8)	1		2.3
A Jugar (Facultad 9)	0		0
Operación Verdad (Facultad 9)		X	0
Control y Gestión de Indicadores (Facultad 9)	0		0
Nova (Facultad 10)	7		16.6
Archivo General de la República de Venezuela (Facultad 10)	0		0



Intranet de PDVSA (Facultad 10)	X	0
---------------------------------	---	---

Tabla 2.8 Productos destinados al beneficio de una o varias organizaciones

Institución	Total	%
La UCI	5	11.9
Organizaciones cubanas que necesitan software para desarrollarse	20	47.6
Organizaciones extranjeras que necesitan software para desarrollarse	24	57.1
Otros	1	2.3

Tabla 2.9 Calidad de los productos que se fabrican en la UCI

	Excelente	Buena	Regular	Mala
	7	22	1	0
%	16.6	52.3	2.3	0
<b>No responde</b>				12 – 28.5%



Tabla 2.10 Evaluación de los estudiantes y profesores según su definición del concepto de calidad

<b>ABIERTA: TRANSCRIBIR TEXTUALMENTE</b>				
<b>Transcribir concepto de calidad:</b>				
Las características de un producto o servicio que le				
-----				
confiera amplitud para satisfacer las necesidades				
-----				
establecidas.				
-----				
	<b>Bien</b>	<b>Regular</b>	<b>Mal</b>	<b>Nc</b>
	16	12	10	4
<b>%</b>	38.0	28.5	23.8	9.5

Tabla 2.11 Equipo de aseguramiento de la calidad por proyectos

	<b>Sí</b>	<b>No</b>	<b>No sé</b>
	27	9	6
<b>%</b>	64.1	21.4	14.2

Tabla 2.12 Personal dedicado a la Gestión de Configuración por proyectos

	<b>Sí</b>	<b>No</b>	<b>No sé</b>
	17	6	19



%	4.04		14.2		45.2
<b>PREGUNTA ABIERTA: TRANSCRIBIR TEXTUALMENTE</b>					
<b>Transcribir concepto de GC:</b>					
Es una disciplina de la IS que se preocupa de:					
-----					
<ul style="list-style-type: none"> <li>• Identificar y documentar las características funcionales y</li> </ul>					
-----					
físicas de los elementos de configuración.					
-----					
<ul style="list-style-type: none"> <li>• Controlar los cambios a tales características.</li> </ul>					
-----					
<ul style="list-style-type: none"> <li>• Reportar el proceso de tales cambios y su estado de</li> </ul>					
-----					
implantación.					
-----					
	<b>Bien</b>	<b>Regular</b>	<b>Mal</b>		<b>Nc</b>
	6	9	5		22
%	14.2	21.4	11.9		52.4



Tabla 2.13 Proyectos que poseen el rol de Administrador de Configuración

	<b>Sí</b>	<b>No</b>	<b>No sé</b>	
	11	9	22	
<b>%</b>	26.1	21.4	52.4	
<b>Diga tres tareas fundamentales de ese rol</b>				
	<b>Bien</b>	<b>Regular</b>	<b>Mal</b>	<b>Nc</b>
	5	3	2	32
<b>%</b>	11.9	7.1	4.7	76.1

Tabla 2.14 Plan de Gestión de Configuración aplicado en cada proyecto

	<b>Sí</b>	<b>No</b>	<b>No sé</b>
	7	10	25
<b>%</b>	16.6	23.8	59.5

Tabla 2.15 Identificación y gestión de las versiones existentes de los programas y

la documentación por proyecto

	<b>Bien</b>	<b>Regular</b>	<b>Mal</b>	<b>Ns/Nc</b>
	9	8	8	17
<b>%</b>	21.4	19.0	19.0	40.4



Seleccione las herramientas que utiliza				
	Visual Source Safe	Subversion	Clear Case	Otras
	5	27	1	9
%	11.9	64.2	2.3	21.4

Tabla 2.16 Control de los cambios antes y después de que el software sea distribuido

al cliente

	Bien	Regular	Mal	Ns/Nc
	4	6	10	22
%	9.5	14.2	23.8	52.3

Seleccione las herramientas que utiliza				
	Visual Source Safe	Subversion	Clear Case	Otras
	5	15	0	5
%	11.9	35.7	0	11.9

Tabla 2.17 Logros que se obtienen con la disciplina de Gestión de Configuración

Actividades	Total	%
Mayor satisfacción de los clientes	25	59.5
No influye en la satisfacción de los	5	11.9



clientes		
Menor satisfacción de los clientes	0	0
Mayor organización y planificación del trabajo.	30	71.4
No influye en la organización y planificación del trabajo.	0	0
Menor organización y planificación del trabajo.	0	0
Aumenta el tiempo que los estudiantes dedican a la fabricación de software.	6	14.2
No influye en el tiempo que los estudiantes dedican a la fabricación de software.	5	11.9
Disminuye el tiempo que los estudiantes dedican a la fabricación de software.	15	35.7
Aumenta la cantidad de clientes	14	33.3
No influye en la cantidad de clientes	12	28.5
Disminuye la cantidad de clientes	0	0
Ns/Nc	11	26.1

Tabla 2.18 Importancia de la Gestión de Configuración según cada proyecto

	Sí	No	Ns/Nc
	33	0	9
%	71.5	0	21.4



<b>Mencione alguna de sus ventajas</b>				
	<b>Bien</b>	<b>Regular</b>	<b>Mal</b>	<b>Ns/Nc</b>
	9	12	4	17
<b>%</b>	21.4	28.5	9.5	40.4

Tabla 2.19 Disciplina de Gestión de Configuración en las facultades

	<b>Sí</b>	<b>No</b>	<b>No sé</b>
	11	5	26
<b>%</b>	26.1	11.9	61.9

### 2.3 Conclusiones arrojadas según el procesamiento de encuestas. Análisis de algunas preguntas significativas.

Entre las diez facultades a un total de 26 proyectos productivos, se aplicaron 42 encuestas. De las cuales la mayoría se destinaron al líder y un estudiante por proyecto. Para poder definir las categorías de excelente, bien, regular y mal, se siguió el método de la revisión de la bibliografía, cada pregunta que se realizaba tenía un fundamento en algún documento que permitiera evaluar la respuesta dada por el encuestado.

El criterio que se utilizó para sacar el por ciento en cada pregunta fue la regla de tres, es decir, la cantidad de encuestas reales de todos los proyectos representa el 100 %, la cantidad de preguntas respondidas representa la variable, para obtener la misma se multiplica la cantidad de preguntas respondidas por el 100% y se divide por la cantidad de encuestas reales de todos los proyectos, por ejemplo:

La cantidad de encuestas reales de todos los proyectos  $\longrightarrow$  100%



La cantidad de preguntas respondidas  $\longrightarrow$  X

$X = \frac{\text{La cantidad de preguntas respondidas} \times 100\%}{\text{La cantidad de encuestas reales de todos los proyectos.}}$

A continuación se presentan las gráficas y las conclusiones que muestran las divergencias en los resultados obtenidos de las preguntas más importantes referidas al proceso de calidad y la disciplina de GC.

Según la mayoría de los encuestados la calidad de sus productos fabricados es excelente y buena para un total de 68.9 % (Fig.2.2), sin embargo se puede apreciar que existe un notable desconocimiento de los elementos y las características que integran el concepto de calidad. Este problema representa el 62 % del total y solamente el 38 % respondió bien (Fig. 2.3).

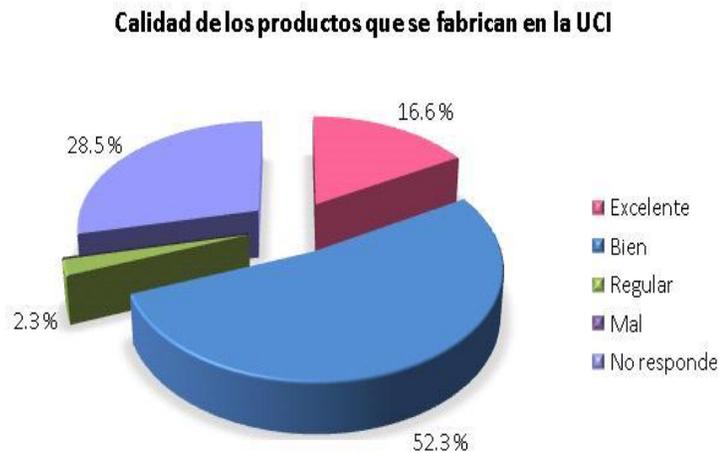


Figura 2.2 Calidad de los productos que se fabrican en la UCI

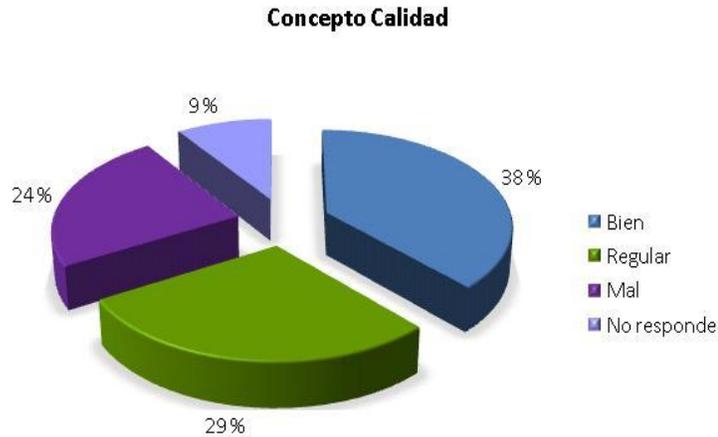


Figura 2.3 Evaluación de los estudiantes y profesores según su definición del concepto de calidad

El 64 % de los encuestados expresaron que en sus proyectos productivos cuentan con un equipo de aseguramiento de la calidad (Fig.2.4), sin embargo solamente el 4.0 % de ellos responden que tienen un personal dedicado específicamente a la GC (Fig.2.5), lo que evidencia el mal manejo de la distribución de los roles dentro del mismo equipo de trabajo, y el desconocimiento de la existencia y utilidad que puede brindar el rol de Administrador de Configuración.

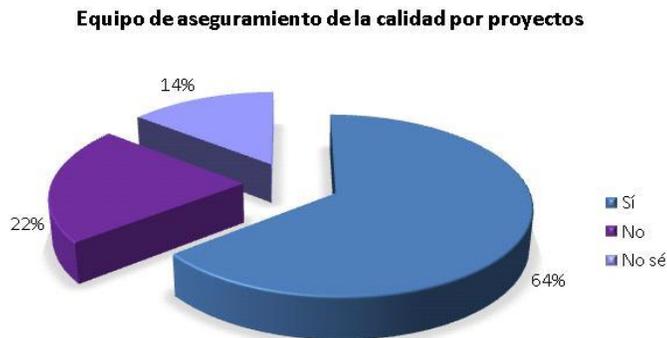


Figura 2.4 Equipo de aseguramiento de la calidad por proyectos



Figura 2.5 Personal dedicado a la Gestión de Configuración por proyectos

En esta gráfica se evidencia la falta de información que existe entre los proyectos productivos y las direcciones de las facultades, solamente el 26.1 % de ellos conocen que en su facultad se controla la GC y cómo se lleva a cabo la misma. La mayoría desconoce totalmente del tema, lo que representa el 61.9 %. Solo el 11.9 % responde que no existe nada de GC en sus facultades, Fig. 2.6.



Figura 2.6 Disciplina de Gestión de Configuración en las facultades



Los resultados obtenidos revelan un grado de desconocimiento sobre el tema en cuestión, en cuanto a la forma de aplicación y distribución de roles, además de destacar la necesidad de estudio preliminar antes de enfrentarse a un proyecto productivo, puesto que existen errores a la hora de identificar conceptos y situaciones determinadas.

Los principales conceptos desconocidos como se demostró anterior mente son: el de calidad del software, el de GC, el de los roles relacionado con esta disciplina. También llama mucho la atención la desinformación que existe entre las direcciones de las distintas facultades y los estudiantes que integran los proyectos productivos.

## **2.4 Ejemplos prácticos de la Gestión de Configuración en la UCI.**

Realizando una entrevista a los líderes y estudiantes de cuatro proyectos de gran alcance de la universidad se estableció un criterio de acuerdo a la utilización de la GC en los mismos, Anexo 4. Los proyectos analizados fueron: SCADA de PDVSA, R&N, Identidad y SIGEP.

Estos proyectos en general no tienen bien definido un modelo de GC por el cual regirse, solamente aplican algunas de las actividades del Proceso de GC, como son: Identificación de los Elementos de Configuración (IEC), Control de Versiones (CV), Control de Cambios (CC) y Generación de Informes del Estado (GIE).

En los cuatro proyectos, dentro de la IEC, se tienen en cuenta las actividades a desarrollar en cada iteración y los artefactos que generaba cada rol dentro del proyecto. El seguimiento a los Elementos de Configuración del Software (ECS) constituye una parte importante dentro esta actividad, concluyendo en la investigación realizada que esta tarea no se ejecuta. Algunos ejemplos de cómo se debe llevar a cabo los ECS aparecen en el Anexo 5.[Febles, 2004]

En el CV se apoyan fundamentalmente en herramientas como el Subversion y el Visual Source Safe para gestionar las versiones de los objetos de configuración creados durante el proceso de desarrollo del software. En esta actividad es importante señalar la falta de comunicación que existe entre los diferentes



módulos dentro de un mismo proyecto, puesto que un cambio en uno de ellos requiere del conocimiento de los demás, debido a que están estrechamente relacionados y se trabaja con la misma documentación.

El CC es una de las actividades fundamental dentro de la disciplina de GC. En los proyectos que se están analizando sí se realiza esta actividad, pero en la mayoría de los casos no aplican todos los pasos que requiere la misma, por ejemplo: no existe una Autoridad de Control de Cambio (ACC), una persona o grupo que toma la decisión final del estado y la prioridad del cambio, y que presente un informe con los resultados de las evaluaciones realizadas.

Para cambios aprobados se debe generar una Orden de Cambio de Ingeniería (OCI), la cual no se realiza con toda la rigurosidad que lleva, principalmente porque se tiende a ser informal y no se acostumbra a documentar todos los cambios realizados.

Entre los cuatro proyectos que se están estudiando, solamente uno de ellos (SCADA de PDVSA), tiene definido algo sobre las Revisiones Técnicas Formales (RTF). Aunque esta debería ser una actividad de obligatorio cumplimiento para todo proyecto productivo; puesto que la identificación, el CV y el CC ayudan a un equipo de desarrollo de software a mantener un orden que llevaría a una situación caótica y sin salida. Sin embargo, los mecanismos más correctos de CC hacen un seguimiento a un cambio solo hasta que se ha generado la OCI. Por lo que se debe llevar a cabo una RTF para cualquier cambio que no sea trivial.

Al llevar a cabo la actividad de GIE, se realiza principalmente para ser mostrada al cliente y no con la funcionalidad de la misma. La GIE desempeña un papel vital en el éxito del proyecto de desarrollo de software. Cuando aparece involucrada mucha gente, es muy difícil que se dé el síndrome de que “la mano izquierda ignora lo que hace la mano derecha”. Un equipo de IS emplea meses de esfuerzo en construir un software a partir de unas especificaciones de hardware obsoletas. Puede que la persona que descubra los efectos secundarios serios de un cambio propuesto no está enterada de que el cambio se está realizando. El IEC ayuda a eliminar esos problemas, mejorando la comunicación entre las personas involucradas.



Un modelo de GC es mucho más que la definición y aplicación de procesos, por lo que es importante señalar que falta mucho camino por recorrer en este sentido. Los cuatro proyectos analizados han encontrado parte de la solución a este problema dándoles la responsabilidad a estudiantes de quinto de año para que en sus trabajos de diplomas desarrollen un modelo de GC que se adapte a las necesidades de cada proyecto.

Están teniendo mucho avance al respecto, pero en el que más se evidencia es en SCADA de PDVSA. Este ha definido un Sistema de Procesos (SP) con todas sus actividades fundamentadas y algunas métricas para de alguna forma relatar las medidas individuales sobre algún aspecto, (por ejemplo: el número medio de errores encontrados por revisión o el número medio de errores encontrados por persona y hora en revisión).

Contribuyendo a estas mejoras, las autoras proponen que no se debe prescindir del rol que plantean diferentes autores para la GC: "Administrador de Configuración". Este será el responsable ante el Jefe del Proyecto de definir e identificar los elementos de configuración a administrar. Además, en este contexto es responsable de controlar la liberación, control de los cambios, presentación de informes sobre las líneas base, y estados de avances a petición del jefe del proyecto. En resumen, es el administrador de la GC del sistema, el cual por definición debe tener un conocimiento pleno de la configuración en su total extensión. [Franco, 2006]

Las actividades fundamentales desarrolladas por este rol son:

- Establecer Jerarquía Preliminar del Software
- Definir Esquema de Identificación
- Establecer y administrar línea base
- Identificación de elementos de configuración
- Control de cambios en la configuración
- Control de versiones
- El lanzamiento de versiones del producto. [Franco, 2006]



### **Conclusiones parciales**

- En la UCI existe un gran desarrollo en la producción del software. Teniendo logros significativos en la comercialización de sus productos para la informatización del país, de la vida de la propia universidad, y la exportación.
- En la UCI todavía no se ha logrado la madurez requerida para informatizar el país, se necesita una buena preparación del personal, desarrollar el software, brindar soporte, crear una infraestructura adecuada, mejoras continuas, etc.
- Se necesita una organización que garantice mejoras continuas de sus soluciones.
- Una decisión importante fue la de diseñar un cuestionario guía con un lenguaje simple para el usuario.
- Existe en los proyectos productivos de la UCI, gran desconocimiento del tema calidad y más puntualmente sobre el proceso central en la producción del software, la GC.
- No tienen definido un Modelo de GC por el cual regirse. Solamente trabajan algunas actividades dentro de los Procesos de GC.

El desafío fundamental para los proyectos productivos está en asumir con seriedad el problema y realizar el mejoramiento de los procesos teniendo un respaldo organizacional que incluya los recursos necesarios.



### **Capítulo III: Modelo de Referencia para la Gestión de Configuración (ModelConfig.UCI)**

La GC es un proceso imprescindible en la producción de software, está escasamente formalizado y ejecutado en la UCI. En el Capítulo se presenta un modelo de referencia para la GC, tomando como punto de partida las buenas prácticas del Sistema de Procesos existente en algunos proyectos productivos e incorporándoles las que se consideran necesarias, para obtener el modelo que satisfaga las necesidades actuales del desarrollo de software.

Se describe la estructura general del modelo, se plantean los procesos y métricas que integran el Sistema de Procesos y el Sistema de Métricas del modelo. Se definen formalmente las métricas y se presentan los procedimientos de los sistemas correspondientes. Además de estos dos sistemas el modelo consta de un Sistema Informático para automatizar algunos de sus componentes.

#### **3.1 Análisis de la Gestión de Configuración en cuatro proyectos productivos.**

En las 42 encuestas aplicadas a 26 proyectos productivos de la universidad se pudo comprobar que solamente el 4% tiene un personal dedicado a la GC, el 16,6% pone en práctica un Plan de GC y el 14,2 % de los encuestados respondió satisfactoriamente el concepto de GC. Estos datos demuestran el desconocimiento sobre esta disciplina tan importante para el desarrollo de un software.

Del pequeño porcentaje que pone en práctica un Plan de GC se hizo una selección de cuatro proyectos productivos (identidad, R & N, SIGEP, SCADA de PDVSA), a los que se les realizó una entrevista para indagar sobre cómo aplican este plan, y de su experiencia tomar las buenas prácticas. De esta selección y añadiéndole nuevas actividades, procesos y sistemas, es que se obtendrá un Modelo de Referencia ajustado a las características y necesidades de la universidad.

Estos proyectos solamente tienen definido algunas actividades dentro de lo que se denomina “Sistema de Procesos” [Febles, 2004], [Pressman, R. S], por lo que no se puede decir que tienen un modelo de GC por el cual regirse. Algunas de estas actividades son:

- Identificación de los Elementos de Configuración (IEC)



- Control de Versiones (CV)
- Control de Cambios (CC)
- Generación de Informes del Estado (GIE)

Es muy importante destacar que a pesar de cumplir con algunas actividades dentro del Sistema de Procesos no significa que estos equipos de trabajo apliquen durante el ciclo de vida de sus proyectos una buena GC, esto solamente les permite tener un control de las revisiones y los cambios que van haciendo a diario, omitiendo todo lo referente al control de las métricas y la actualización de los cambios que deben realizar en las herramientas usadas para este proceso.

### **3.2 Modelo conceptual para la definición de ModelConfig.UCI**

En la Fig. 3.1 se muestra la estructura básica para elaborar el modelo ModelConfig.UCI (Modelo de Referencia para la GC en la UCI).

El modelo está compuesto por tres sistemas:

Sistema de Procesos

Sistema de Métricas

Sistema Informático

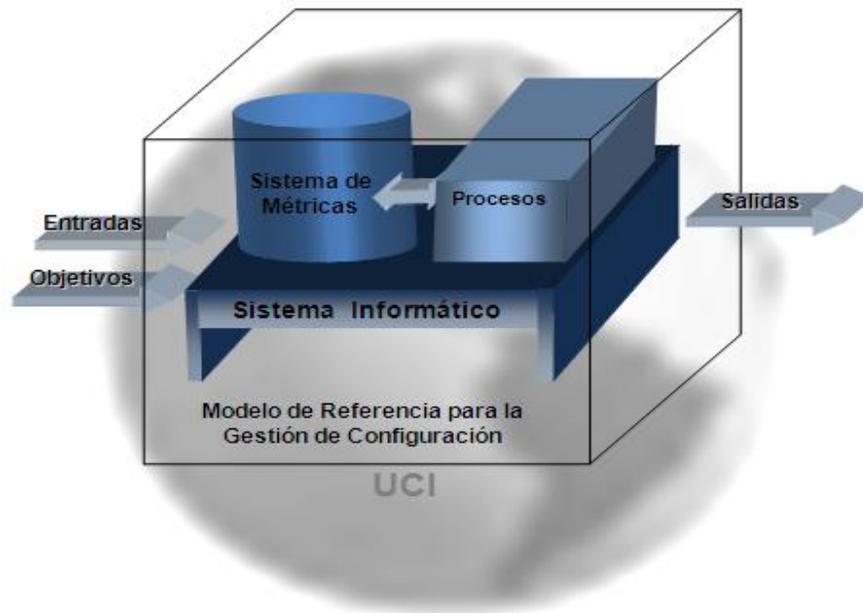


Figura 3.1 Modelo conceptual de ModelConfig.UCI

Los tres están estrechamente vinculados. Las métricas se evalúan a partir de los datos que se obtienen con la ejecución de los procedimientos del Sistema de Procesos y en muchos casos los datos utilizados por este sistema están compuestos por los resultados de las mediciones realizadas en el Sistema de Métricas. También la posibilidad de mejorar de forma continua los procesos será posible a partir de los resultados de las métricas aplicadas a estos.

Como soporte a estos dos sistemas, se propone un Sistema Informático que garantizará eficiencia en la ejecución de los procesos y rapidez en el tratamiento de la información. El sistema Informático, está compuesto por varias herramientas CASE (**Computer Aided Software Engineering**), que permitirán ejecutar los dos sistemas iniciales de forma automatizada, sirviéndose de los beneficios de utilizar la tecnología en la ejecución de procesos. [Hernández, 1996]

Del examen de varias definiciones de GC descritas en el Capítulo 1 se establecen para el modelo un conjunto de entradas, salidas y objetivos básicos; estos son:



a) Objetivos

Establecer y mantener la integridad de los productos de Software generados durante un proyecto.

Evaluar y controlar los cambios sobre estos productos.

Facilitar la visibilidad y la información sobre el proyecto.

b) Entradas

Componentes de un proyecto de software

Solicitudes de cambios

Metas del proyecto

c) Salidas

El Plan de Gestión de la Configuración.

La Identificación de la Configuración.

Estado de cada Elemento de Configuración de Software (ECS)

Los Informes del estado de la configuración.

### **3.3 Sistema de procesos incluido en ModelConfig.UCI**

Como fue descrito en el Capítulo 1, son propósitos de los procedimientos de un Sistema de GCS asegurar que el estado de los productos asociados al proyecto sea conocido, que éstos estén disponibles todo el tiempo y que a las acciones realizadas sobre ellos pueda dársele seguimiento a través de todo el ciclo de vida del proyecto. [Humphrey, 1989].



Para el desarrollo de la investigación se analizaron y redefinieron los procesos asociados a la GC según las fuentes consultadas. Por lo que se propone un sistema de cinco procesos que tienen que implantarse en la UCI para realizar de forma madura la GC y que son incluidos en el modelo de referencia que se obtiene como resultado de la investigación (Fig. 3.2).

Para hacer la propuesta se tuvieron como premisas las especificaciones de la IEEE, ISO y de CMM, vistas en el Capítulo 1 y la documentación teórica que se ha publicado sobre el tema, [Ginsberg, 1995] [Paulk, 1998] [Antonio, 2001] basándose en las características del dominio de aplicación que ha servido de base a la investigación desarrollada y la experiencia de diferentes especialistas en el tema, tanto en la actividad docente como en la permanente interacción con los objetos del dominio, mediante la aplicación de herramientas a individuos y colectivos. Los procesos seleccionados para ModelConfig.UCI son los siguientes:

- Identificación de los Elementos de Configuración (**IEC**)
- Gestión de Cambios en la Configuración (**GCC**)
- Gestión de Versiones (**GVE**)
- Planificación de la Gestión de Configuración (**PGC**)
- Generación de Informes y Comunicación del Estado de la Configuración (**GIE**)

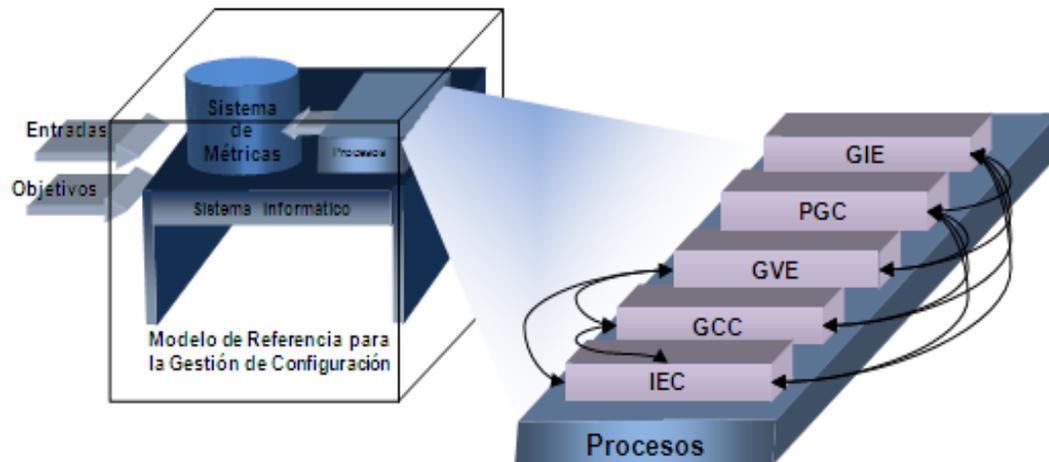


Figura 3.2 Sistema de procesos de ModelConfig.UCI

El tratamiento de conjunto de estas cinco actividades dentro de un Sistema de GC es bastante novedoso. La propuesta elegida en el presente trabajo coincide con la de [Febles, 2004], pues la actual investigación es continuidad de la desarrollada por dicha autora, adaptada a las condiciones y necesidades de los productos fabricados en la UCI.

Las áreas de procesos **IEC**, **GCC** y **GIE** son coincidentes en los tres estándares analizados ISO, IEEE y CMM. [Villamil, 1999] [IEEE, 1990] [ISO, 1995] [Paulk, 1993] [Antonio, 2001] Por su importancia e incidencia dentro del proceso fueron seleccionados para formar parte del modelo por las siguientes razones:

Es importante que se establezca en cada proyecto un método único para la identificación de los EC y que sea conocido por todas las partes involucradas, para que todos los integrantes del mismo se comprometan con sus objetivos y actúen uniformemente.

Cuando se construye un software los cambios son inevitables. Una de las principales amenazas para la calidad de software viene justamente de los cambios, fuente aparentemente benigna. No atender, analizar y ejecutar los cambios de forma organizada implica, entre otros, problemas de calidad, de satisfacción del cliente, pérdida de información, etc. [Pressman, 1997]



Una de las reglas de oro de CMM es, “mantenga informado a todos los interesados”. La generación y transmisión de informes es una manera de cumplir con esta regla.

Para el proceso **PGC** se cree necesario hacer una acotación con respecto al propuesto por [Febles, 2004] puesto que se considera parte de la propuesta: “por la importancia de la planificación para organizar tareas en tiempo, destinar recursos, asignar responsabilidades y posteriormente poder darle seguimiento a lo planificado. Para el control de la configuración es clave, pues es una actividad que se realiza durante todo el ciclo de vida del proyecto, incide en todas las componentes del proyecto e involucra a todos los miembros del equipo de trabajo”.

Las autoras del presente coinciden con este planteamiento, más consideran no obstante, que la planificación de tareas, no es una actividad propia de la GCS, sino de la Gestión del Proyecto, aunque sí presenta un fuerte vínculo con la GCS. Preferirían por tanto, limitar a la PGC a las actividades descritas en [Antonio, 2001] [Rational, 2003] que atribuyen a la PGC, como salida principal, el artefacto Plan de Gestión de Configuración, el cual contiene, entre otros, la planificación de elementos organizativos, responsabilidades, hitos en la implementación del plan, políticas, directivas y procedimientos a aplicar durante la GCS. Se planifica además cómo se ejecutarán las distintas actividades de la GCS que se realizarán en el proyecto, se especifican los períodos para la recolección y almacenamiento de registros, es decir, qué documentación almacenar, métodos para dicha recolección y período de tiempo para hacerlo.

Como puede observarse, en estos elementos, no hay espacio para la planificación de tareas del proyecto, asignación de recursos humanos o materiales a actividades, sino que esto sigue siendo parte de la planificación del proyecto efectuada como una actividad de gestión de proyecto y no de GCS.

En el caso del proceso para la Gestión de Versiones (**GVE**), es referido indistintamente por algunos autores pero no es incluido como un proceso independiente en ninguno de los estándares analizados. [Antonio, 2001] [Ginsberg, 1995] [Gervás, 2002] Se decidió incluirlo por las siguientes razones:



Dentro del control de configuración los dos elementos fundamentales son los cambios y las versiones asociadas a los ECS. Ninguno puede ser obviado en la búsqueda de eficiencia y calidad. El proceso de Gestión de cambios ya fue incluido en el modelo.

Cuando se mezcla el control de versiones con el control de los cambios en un proceso único, sin delimitar fronteras y actividades específicas se diluyen las responsabilidades y no se ejecutan muchas tareas que son imprescindibles como la actualización de los repositorios, el control de acceso a los ECS, el seguimiento de componentes que no son ECS, etc.

Un proceso que es incluido en IEEE, CMM e ISO, es el de Auditoría de la Configuración. La auditoría es la forma de comprobar que, efectivamente, el producto que está en construcción es lo que pretende ser. Este proceso no ha sido incluido en el modelo que se propone, pues se considera que para los proyectos productivos en la UCI es mucho más factible y viable ubicar la auditoría de configuración fuera de la GC y dentro de la Garantía de Calidad llevada por un grupo externo al proyecto. Las actividades asociadas con la auditoría son extremadamente costosas, requiere de personal experimentado, y con un gran conocimiento del proceso de desarrollo. Sin embargo, debe ser realizada por personal ajeno al equipo de desarrollo técnico para mantener la objetividad de la auditoría. Lograr esto individualmente en cada proyecto en la UCI no es factible. Por el número reducido de especialistas con que cuenta solo podrá ejecutarse si centran todas las auditorías en un único grupo que sea externo al proyecto que será auditado.

Las auditorías que se realizan habitualmente son:

**Auditoría Funcional:** Cuyo objetivo es comprobar que se han completado todas las pruebas necesarias para el / los ECS auditados, y que, teniendo en cuenta los resultados de los tests, se puede afirmar que el / los ECS satisfacen los requisitos que se impusieron sobre él.

**Auditoría física:** Cuyo objetivo es verificar la adecuación, integridad y precisión de los elementos físicos de documentación que constituyen la Línea Base. [Jorge, 2003]



Lo anteriormente expuesto no quiere decir que las ACS estén desvinculadas de la GC, pues ¿ cómo es que se puede asegurar que el cambio se ha implantado correctamente?, la respuesta es doble: mediante las Revisiones Técnicas Formales y las Auditorías de Configuración de Software [Pressman, R. S]

Las autoras proponen separar las RTF del las ACS, es decir, las ACS las realizaría un grupo externo al proyecto por su complejidad y las RTF se aplicarían a nivel de proyecto. El hecho de separar este mecanismo de control no quiere decir que estén independizadas una de la otra, puesto que las ACS completan las RTF al comprobar características que generalmente no tiene en cuenta la revisión. Esto se plantea de esta forma pues en el modelo que se está proponiendo la GC es una actividad formal, y por tanto la ACS se debe llevar a cabo independientemente.

Las RTF se centran en la corrección técnica del EC que ha sido modificado. Es realmente una clase de revisión que incluye recorridos, inspecciones, revisiones cíclicas y otro pequeño grupos de evaluaciones técnicas del software. Cada RTF se lleva a cabo mediante una reunión y solo tendrá éxito si es bien planificada, controlada y atendida [Pressman, R. S]

Las inspecciones pueden ser vistas como una implementación de las revisiones formales del software las cuales representan un filtro para el proceso de IS, éstas se aplican en varios momentos del desarrollo y sirven para detectar defectos que pueden así ser eliminados [Freeman, 1990]. Los inspectores, deben en primer lugar tener acceso a los elementos a inspeccionar, los cuales no son otra cosa que ficheros, es decir, ECS. Esta necesidad impone que el sistema empleado para la GCS, permita el acceso a los elementos a inspeccionar; se ha encontrado otro punto de contacto de la GCS con otras actividades, en este caso las inspecciones al software. Incluso la relación con esta actividad es aún mayor, ya que como resultado de las inspecciones, como ya se comentó, se detectan errores y defectos, que implican la necesidad de corregir ECS, por lo que deberán cambiar, lo cual significa que una de las principales salidas del proceso de inspecciones (defectos) constituye entrada de uno de los subprocesos de la GCS (el Control de Cambios) [Franco, 2006]

Se ha comprobado que la acción simultánea de los cinco procesos produce una sinergia beneficiosa para gestionar los ECS en la UCI.



### 3.3.1 Procedimientos y gráficos en el Sistema de procesos

En los enfoques tradicionales de calidad se considera que un proceso se gestiona con calidad cuando se identifica, tiene definidas sus interrelaciones, está documentado, se controla y se mejora. [Yooker, 1997] Un proceso estará institucionalizado cuando esté documentado a través de diagramas, procedimientos, registros y otros. Documentar los procesos permite que todos los involucrados vean con mayor claridad el trabajo a realizar y el rendimiento esperado, hace más fácil el entrenamiento, facilita el seguimiento del proceso y posibilita mayor satisfacción del cliente. [Michelena, 2001] [Haque, 1995]

Las autoras proponen la utilización del formato propuesto en [Febles, 2004] para documentar los procesos de ModelConfig.UCI. Este formato incluye: Objetivo, Establecimiento de los Roles, Entradas, Definiciones, Actividades del procedimiento, Salidas y Gráficos.

Al considerarse que los diagramas de actividades del Lenguaje Unificado de Modelación (UML) pueden aplicarse al modelado de flujo de trabajo en procesos de negocio, algunos autores han sugerido utilizarlos también para el modelado de los procesos de software. [Oktaba, 2001a] [Oktaba, 2001b].

Este diagrama fue utilizado en el modelo para lograr abstraer y sintetizar los elementos cruciales de los procesos de ModelConfig.UCI en un gráfico. [Rational, 2001] [Rational, 2000] [Jacobson, 2000] [Rumbaugh, 1999] Los gráficos son utilizados como complemento de la descripción de las actividades para facilitar su comprensión de manera gradual e integrada. [Oktaba, 2001a] [Oktaba, 2001b]

En el documento “Guía práctica del modelo de referencia MConfig.pm” (Guia\_MConfig.pm) elaborado por [Febles, 2003] se encuentran los detalles de los procesos incluidos en el modelo. En los epígrafes siguientes se presenta la descripción del proceso de **IEC** y los gráficos asociados la **GCC** y la **GVE**.

### 3.3.2 La identificación de los elementos de configuración

Para la GC, identificar y asignar nombres significativos y consistentes a todos y cada uno de los elementos que forman parte del producto software, es básico. [Antonio, 2001] [Marín, 2002] Tal y como se



justificó en los capítulos anteriores el proceso de identificación de los ECS de software es imprescindible en la GC.

### 3.3.2.1 Consideraciones generales para el proceso

Las actividades en el proceso de IEC fueron seleccionadas por su importancia, implicación en el resto de los procesos y posibilidad de obtener información con la aplicación de las métricas.

Dentro de estas resalta la importancia de la Definición de un Esquema de Identificación. Para el seguimiento del proyecto, la reutilización de componentes y la estimación, entre otros, resulta imprescindible que los componentes se nombren con un esquema de identificación que permita etiquetar y diferenciar los ECS de cada proyecto.

Las autores proponen como un esquema para el almacenamiento de los ECS en el repositorio y para nombrar los componentes del proyecto. La información que aparece en el esquema propuesto, Anexo 5 servirá de referencia para los líderes de proyectos, administrador de configuración, etc. para nombrar los ECS. [Febles, 2004]

Para el establecimiento de la Línea Base pudieran considerarse dos formas:

**FÍSICAMENTE:** Etiquetando cada ECS y almacenándolo en un Archivo o Biblioteca.

**LÓGICAMENTE:** Publicando un documento de identificación de la configuración con el estado del producto en un punto del proceso de desarrollo.

En la investigación se seleccionó la línea base para los proyectos de la UCI de tipo **LÓGICAMENTE** pues de esta manera es más fácil de controlar, seguir y de publicar entre todos los involucrados en una organización pequeña, entiéndase como un proyecto productivo. Utilizarla con el modelo Físico implicaría el establecimiento de una biblioteca que requeriría de una o varias personas dedicadas a su mantenimiento y a mantener informados de esto a todos los involucrados, y requiere un gasto adicional de tiempo, recursos humanos y materiales.



### 3.3.2.2 Procedimiento para la identificación de los Elementos de Configuración del Software

En correspondencia con lo anterior el procedimiento definido para la **IEC** se formaliza como sigue:

**Nombre del proceso:** Marco de Referencia para el proceso **IEC**.

1. Objetivo: Establecer un marco de referencia para la definición e implantación del proceso **IEC**.
2. Establecimiento de Roles: Líder del proyecto, Administrador de Configuración.
3. Entradas: Componentes del proyecto.
4. Definiciones: Estados de las versiones de los componentes del proyecto

**Promoción:** Versión disponible para los desarrolladores

**Estable:** Versión que ya está dentro del control de configuración. (ECS)

**Liberación:** Versión entregada a un cliente (ECS)

5. Actividades en el procedimiento:

Se establece el esquema de identificación de los componentes y los ECS siguiendo el propuesto en MConfig.pm [Febles,2004], Anexo 5.

Se determinan los componentes que serán ECS y cuando pasan al control de configuración.

Se establece las relaciones que van a tener los ECS. Algunas relaciones pueden ser por ejemplo

**Equivalencia:** cuando determinado ECS está almacenado en varios lugares diferentes pero todas las copias corresponden al mismo ECS.



**Composición:** ECS que esté compuesto por otros artefactos de la metodología o de otros módulos del producto.

**Dependencia:** Relaciones entre ECS, fundamentalmente para facilitar el seguimiento de los requisitos.

**Derivación:** A partir del qué se ha originado algo.

**Sucesión:** Historia de los cambios sobre un ECS desde una revisión a otra.

Se **proponen** componentes para convertirse en ECS

Se **autoriza** a que el componente pase a ser ECS

Se **establecen** las relaciones del nuevo ECS con el resto de los ECS del repositorio.

Se **coloca** versión del ECS en el repositorio del proyecto. La versión pasa al estado **estable o base**

Se **actualiza** la Línea Base del proyecto para cada modificación del repositorio.

Se **genera** un documento que contiene la identificación de la configuración, que corresponde al estado actual de la Línea Base en dicho punto del proceso de desarrollo.

Se **publica** este documento y se **distribuye**.

6. Salidas:

Línea base actualizada, esquema de identificación de los ECS, lugar físico para el repositorio del proyecto e informe con la identificación de los ECS

7. Gráfico del proceso:

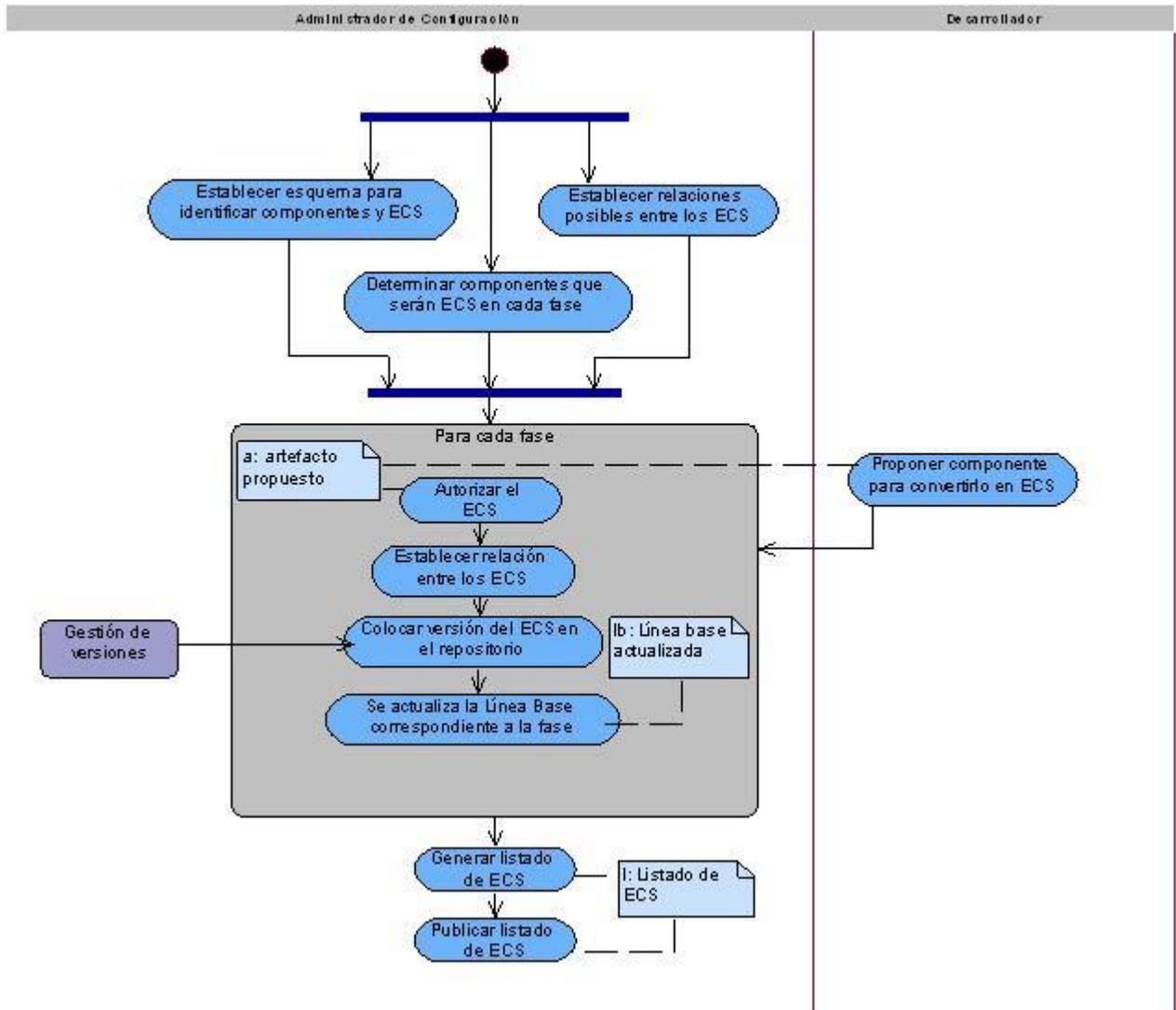


Figura 3.3 Gráfico del proceso de Identificación de los Elementos de Configuración

### 3.3.3 Los cambios a los elementos de configuración

Los cambios durante el transcurso de un proyecto son inevitables. Ocurren, fundamentalmente, porque los involucrados en los proyectos, con el tiempo, incrementan sus conocimientos y aumentan o varían las



necesidades o requisitos. En esta investigación se ha realizado un análisis detallado relacionado con los cambios a los ECS donde se obtuvieron como resultado las primeras aproximaciones de un procedimiento de control de cambios. [Febles, 2000a] [Febles, 2000b] [Febles, 2000c] [Febles, 2002]

### **3.3.3.1 Consideraciones generales para el proceso**

El control de los cambios debe incluir un conjunto de acciones formales y documentadas que define los pasos que deben seguirse para modificar los ECS del proyecto. Dentro de la gestión de los cambios el grado de formalización de la solicitud de cambio varía según los estándares internacionales. Los enfoques tradicionales proponen el proceso como aparece en el Anexo 6.

En estos enfoques, los cambios nuevos que surgen dentro de un desarrollo se tratan de posponer hasta la fase de mantenimiento para no perder el hilo conductor del proyecto. El modelo tradicional es un modelo muy rígido, de tiempo de respuesta lenta y caracterizada por la acumulación de trabajos pendientes que pudieran ser de envergadura.

En enfoques más modernos, como el desarrollo iterativo incremental, se proponen esquemas como el que se muestra en la figura del Anexo 6.

En este enfoque, todos los cambios que surgen dentro de un desarrollo son analizados incluso los más elementales. Los que se posponen pueden no ser ejecutados nunca y el usuario no recibe respuesta inmediata, lo que puede implicar la pérdida de clientes y / o la insatisfacción con el proyecto.

El procedimiento que se propone para el sistema de procesos de ModelConfig.UCI resulta un punto medio entre los modelos tradicionales y los modelos modernos presentados anteriormente (epígrafe 3.4.3.2).

Varios sistemas de control de cambios incluyen un Comité de Control de Cambios (CCC) responsable de la aprobación o el rechazo de las solicitudes de cambio. [Paulk, 1998] [Antonio, 2001] [IEEE, 1990] [IEEE, 1987] [Sorensen, 1999] La estructura de este comité depende del tamaño y la complejidad de los proyectos de la universidad. Para el sistema que se propone se ha decidido distribuir las tareas asociadas a este rol (CCC) entre el administrador de configuración y el jefe de proyecto, haciendo partícipe de



algunas decisiones a estudiantes que dominen temas específicos en el proyecto. En el caso de incluir el CCC implicaría utilizar personas que adicionalmente desempeñan otro rol teniendo que distribuir el tiempo en ambas tareas. Adicionalmente a esto, el modelo que se propone debe ser de fácil comprensión y ejecución para proyectos que estén en fase de inicio.

### **3.3.3.2 Procedimiento para la gestión de los cambios**

Las autoras proponen una formalización del procedimiento de la gestión de los cambios que servirá como marco de referencia para los proyectos productivos. Para definirlo se tuvo en cuenta las características de la producción de software en la UCI, las propuestas de varios autores [Ginsberg, 1995] [Wieggers, 1998] [Antonio, 2001] [Paulk, 1998] [Álvarez, 2000] [Álvarez, 2003] [Febles, 2000a] [Febles, 2000b] [Febles, 2000c] [Febles, 2002] [GTI, 2002] Se ha tenido en cuenta adicionalmente la regla de oro de la gestión de proyectos de: “avise a los interesados lo que está pasando cuanto antes y automáticamente”.

En los diagramas de actividad que aparecen en las Fig. 3.4 y 3.5 se puede observar las actividades, cambios de estados y roles asociados al proceso de Gestión de Cambios a los Elementos de Configuración. [Febles, 2004]

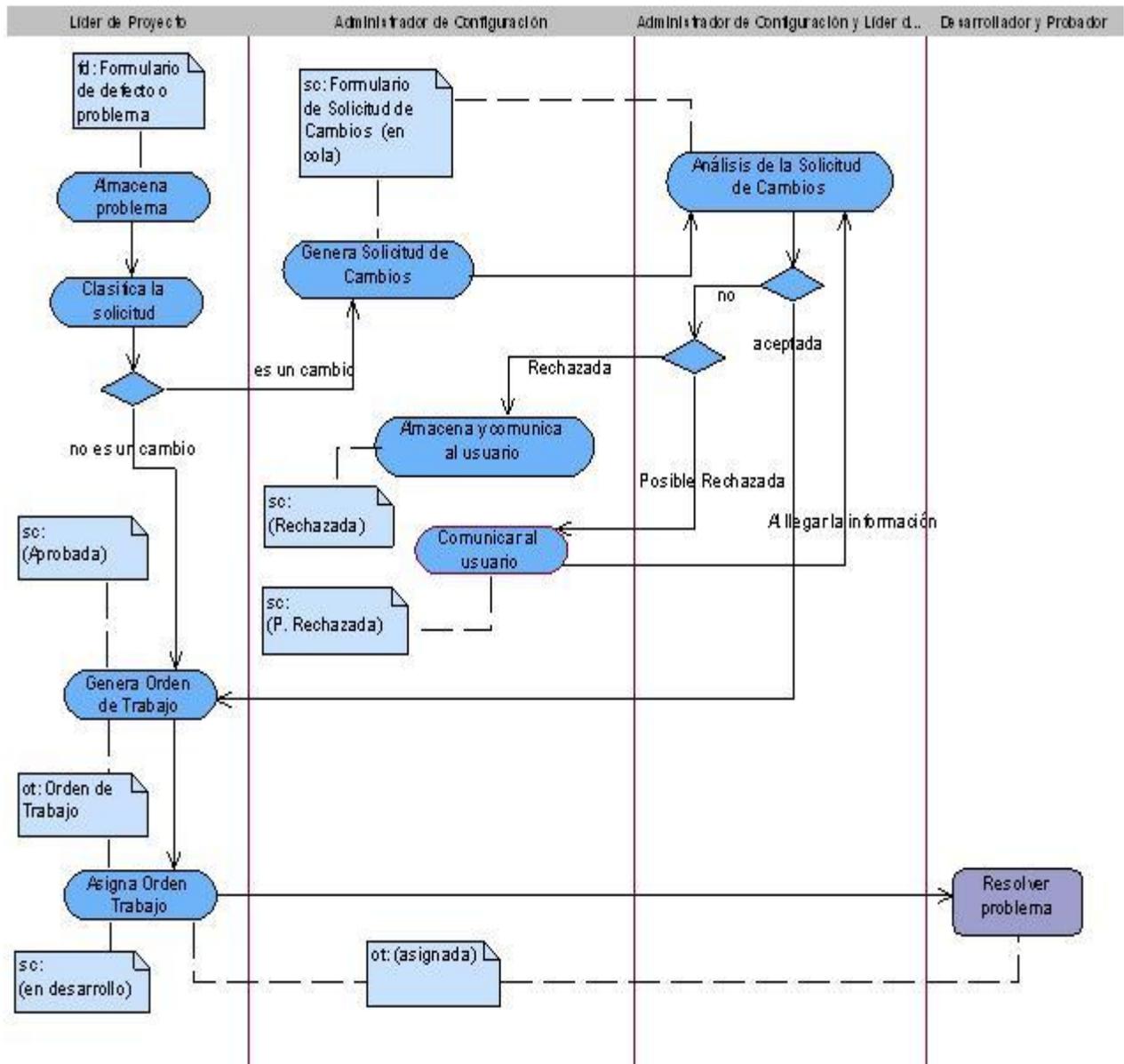


Figura 3.4 Diagrama de actividad # 1 del proceso Gestión de Cambios en la Configuración



Resolver Problema

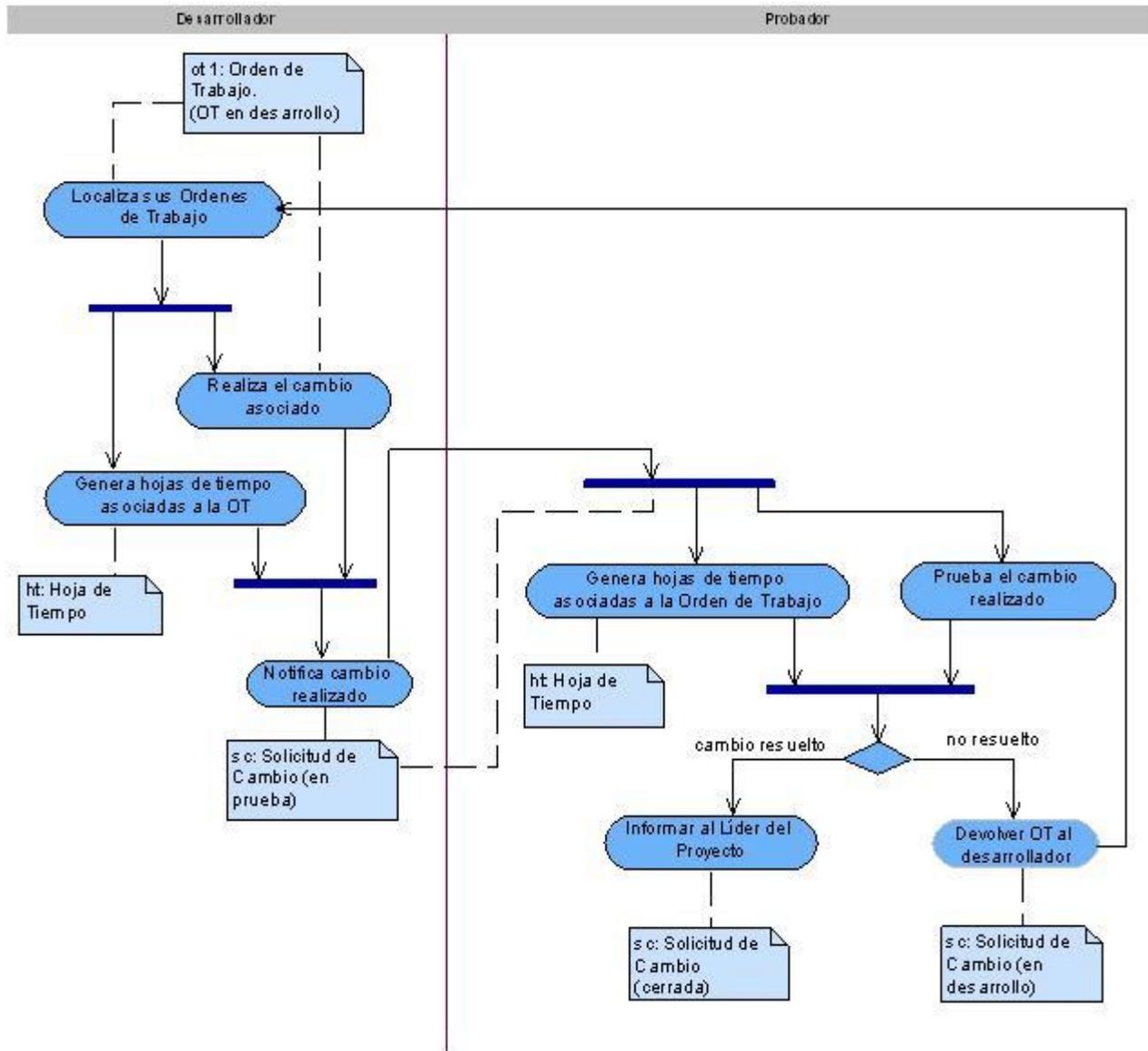


Figura 3.5 Diagrama de actividad # 2 del proceso Gestión de Cambios en la Configuración



### **3.3.4 La planificación en la gestión de la configuración**

El Plan de Gestión de Configuración es un documento que se produce al comienzo de cada proyecto y que define las políticas, estándares y procedimientos que se van a utilizar para gestionar la configuración en el transcurso de dicho proyecto. [IEEE, 1990]

En el Anexo 7 aparece una plantilla que propone la Dr. Ailyn Febles, para documentar el Plan de Gestión de Configuración, adaptada al modelo ModelConfig.UCI. La selección se basó en los datos, estructuración y claridad que tiene el modelo de la IEEE para el documento oficial de la planificación.

### **3.3.5 Las versiones en la gestión de configuración**

Teniendo en cuenta los problemas asociados con el control de versiones en los proyectos productivos debe mantenerse un registro histórico claro y preciso de las diferentes versiones por las que pasan los ECS que permitirá obtener información y/o la recuperación de cualquiera de ellos. Siendo este un proceso indispensable y que requiere de un seguimiento sistemático. [Febles, 2000a] [Febles, 2000b] [Febles, 2000c]

#### **3.3.5.1 Consideraciones generales para el proceso de control de versiones**

Muchos modelos y procesos para el control de versiones determinan que toda modificación conduce a una nueva versión del elemento. [Antonio, 2001] Esta variante implica que el número de versiones crece desmesuradamente siendo compleja la localización y la selección de información, para la UCI resultaría extremadamente costosa y en algunos casos resulta imposible de ejecutar el control de versiones siguiendo este modelo.

En otros modelos se plantea que el usuario define cuando se crea una nueva versión. [Antonio, 2001] Esta variante puede ser utilizada solamente para el producto final, en el caso del resto de los ECS no hay conocimiento del usuario de su existencia ni de los elementos necesarios para la valoración. [Febles, 2000a] [Febles, 2000b] [Febles, 2000c]



En el procedimiento que se propone en este trabajo se establece un punto medio entre ambas variantes.

### **3.3.5.2 Procedimiento para el control de versiones**

Se propone como resultado de la investigación un conjunto de actividades asociadas a un procedimiento para el control de versiones. Con este procedimiento como marco de referencia se establece un modelo que permite controlar de forma ordenada las versiones de los ECS de un proyecto en la UCI.

Los roles que participan en este proceso son: Jefe de proyecto, Administrador de Configuración, Desarrollador y Probador. En su ejecución las versiones transitan por los mismos de las componentes definidas en el proceso de Identificación de los Elementos de Configuración

En los diagramas de actividad que aparecen en las Fig. 3.6 y 3.7 se puede observar las actividades, cambios de estados y los roles asociados al proceso de GVE. [Febles, 2004]

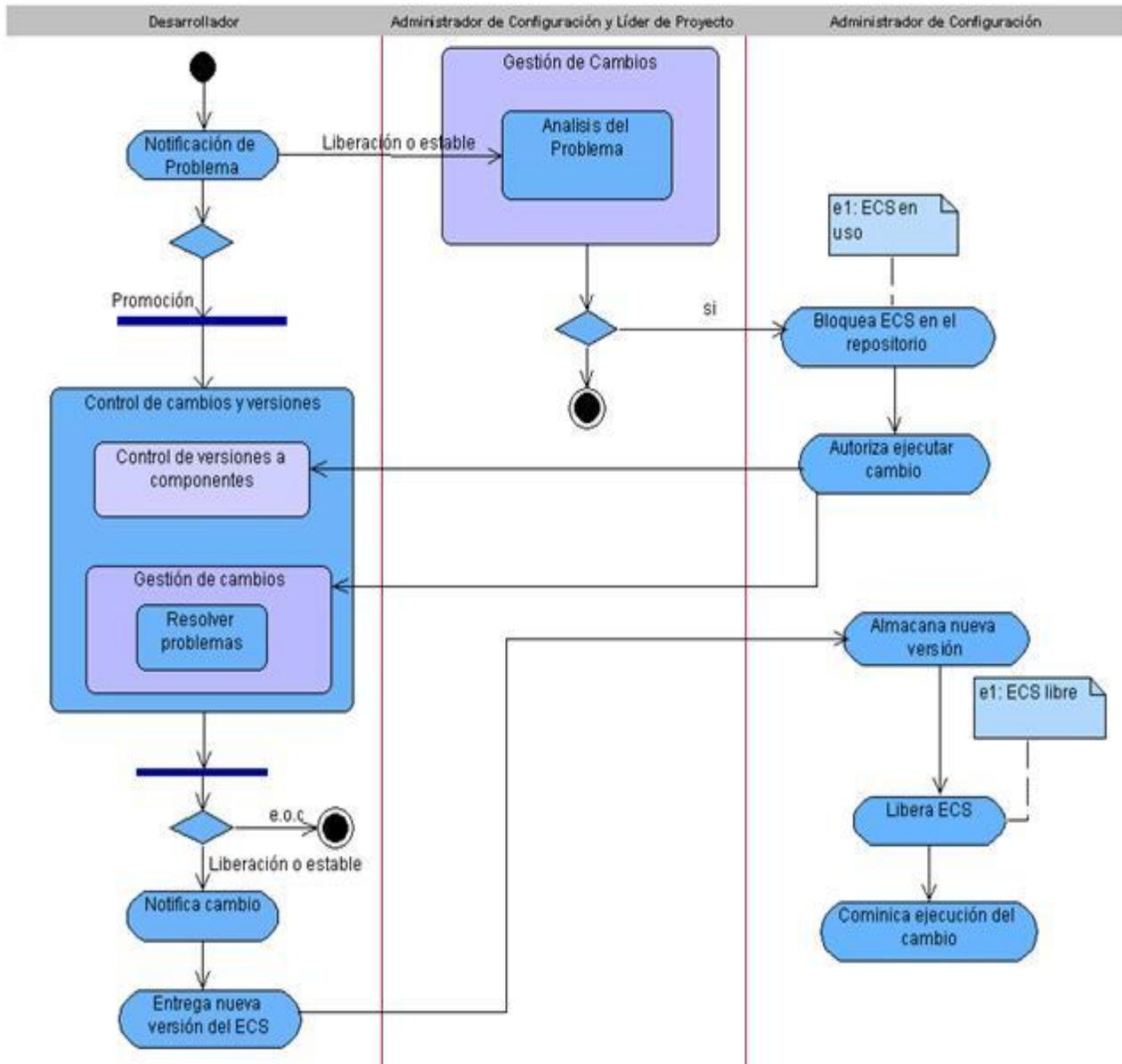


Figura 3.6 Diagrama de actividad # 1 del proceso Gestión de Versiones

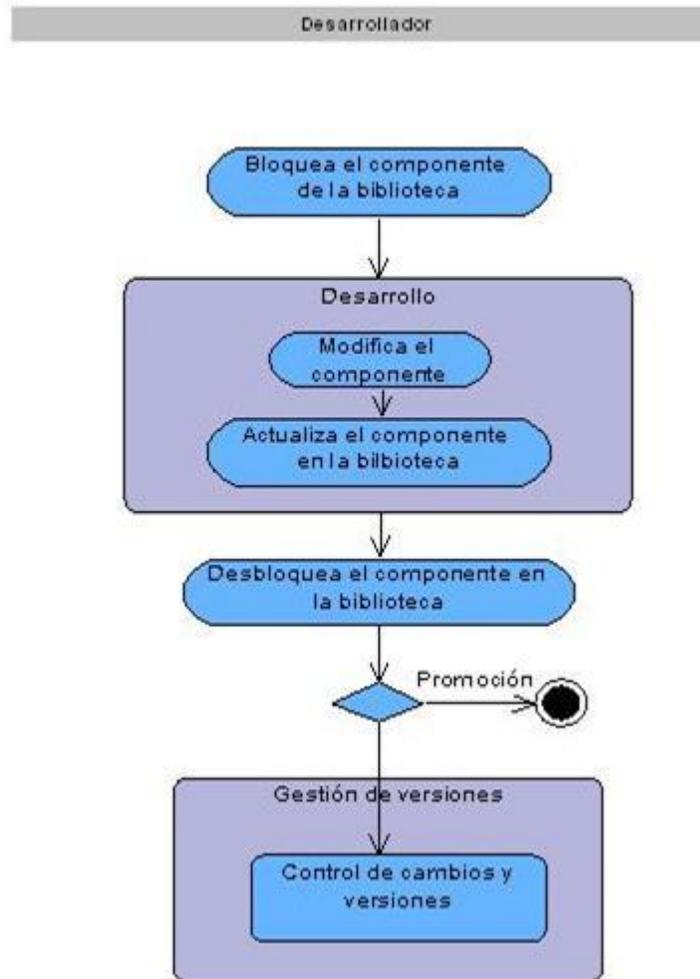


Figura 3.7 Diagrama de actividad # 2 del proceso Gestión de Versiones

### 3.3.6 Relación del Sistema de procesos de ModelConfig.UCI con RUP

El proceso unificado de Rational se repite a lo largo de una serie de ciclos que constituyen la vida de un sistema. Cada ciclo concluye con una versión del producto y se divide en cuatro fases. Dentro de cada



fase se puede descomponer el trabajo en iteraciones que transitan por diferentes flujos de trabajo. Una iteración típica pasa por los cinco flujos de trabajo.

Los procesos descritos en ModelConfig.UCI se insertan en el desarrollo iterativo e incremental del RUP. En cada fase se realizan acciones que deben realizarse siguiendo las especificaciones de los procesos de ModelConfig.UCI relacionadas con los flujos de trabajos específicos. En la tabla 3.1 aparecen reflejados los procesos de ModelConfig.UCI que se realizan en diferentes momentos de intersección entre las fases y las iteraciones del RUP.

Tabla 3.1 Relación RUP-ModelConfig.UCI

Flujos de Trabajo	Planificación, Análisis de Riesgo y Preparación	Elaboración	Construcción y Verificación	Transición
Identificación	IEC , GIC	PGC, IEC	IEC, GCC, GVE, GIE	IEC, GCC, GVE, GIE
Análisis	IEC , GIC	PGC, IEC, GCC, GVE, GIE	IEC, GCC, GVE, GIE	IEC, GCC, GVE, GIE
Diseño	IEC , GIC, PGC	PGC, IEC, GCC, GVE, GIE	IEC, GCC, GVE, GIE	IEC, GCC, GVE, GIE
Implementación		PGC, IEC, GCC, GVE, GIE	IEC, GCC, GVE, GIE	IEC, GCC, GVE, GIE
Prueba		PGC, IEC, GCC, GVE, GIE	IEC, GCC, GVE, GIE	IEC, GCC, GVE, GIE

### 3.4 Sistema de métricas en ModelConfig.UCI

Mantener a los usuarios, a los directivos y a los desarrolladores al tanto del estado de la configuración y su evolución, literalmente, permite dar respuesta a la pregunta “¿Qué ocurrió?”, y también a la pregunta “¿Cuándo ocurrió?” entre otras que aparecen a lo largo del proyecto. [Fernández, 1999]



En la medida en que el proceso de Generación de Informes y Comunicación del Estado de la Configuración (**GIE**) se enriquezca con métricas que permitan transmitir mayor cantidad de información que den respuesta a las preguntas que surgen durante el desarrollo del proyecto, aumenta la posibilidad de tomar medidas correctivas a tiempo y también ayuda en la toma de decisión a los líderes de proyecto, jefes de desarrollo, etc.

Como resultado de la investigación y teniendo en cuenta las sugerencias de Pressman y la Dr. Febles de las métricas para las PYME, se seleccionó un Sistema de Métricas para la GC que forma parte del modelo propuesto [Pressman, 2000] [Febles, 2004]. Este sistema está formado por métricas seleccionadas de **Personal Software Process** (PSP), **Team Software Process** (TSP), **Process Software Measures** (PSM) y otros autores. [Álvarez, 2000] [Álvarez, 2003] [Febles, 2000a] [Febles, 2000b] [Febles, 2000c] [Febles, 2002] [GTI, 2002].

La mezcla de los cinco procesos propuestos en la investigación permite controlar elementos que con las propuestas anteriores (CMM, ISO e IEEE) no era posible hacerlo pues uno de estos procesos no se contemplaba en ninguno de los dos estándares y otro solo se contemplaba en el de la IEEE. A partir de la combinación de estos procesos las autoras proponen en los siguientes epígrafes la utilización de un conjunto de métricas definidas por diferentes autores y adaptadas a las características y necesidades de los proyectos productivos de la UCI.

Las métricas para la GC sirven como complemento a la justificación realizada en epígrafes anteriores de la selección de los cinco procesos incluidos en el modelo y de la combinación de los mismos. Para ello se definen como:

Una métrica es una  $n$ -tupla  $(P, Q, \Phi, M)$ , con  $n = 4$ , donde:

**P** es un conjunto finito de datos que se obtienen en el proceso **PP**

**Q** es un conjunto finito de datos que se obtienen en el proceso **PQ**

**M** es el conjunto de mediciones posibles



$\phi$  es una función definida de la forma siguiente:

$$\begin{aligned} \phi : P \times Q &\rightarrow M \\ \phi, q &\rightarrow \phi, q = m \end{aligned}$$

### 3.4.1 Métricas de la gestión de los cambios y la planificación.

La combinación de los procesos GCC y PGC se sugería por IEEE y CMM, no así por ISO. La existencia de ambos en la GC permite que los informes del estado de la misma se enriquezcan con métricas que son generales al proyecto. Se decidió que a partir de la combinación de los mismos era posible obtener mediciones asociadas con el control del avance del proyecto, los errores de estimación de tiempo y el cumplimiento de la planificación. Como resultado de esta investigación las métricas anteriores son adaptadas para el proceso específico de GC a partir de tener en el sistema propuesto el proceso de planificación de la configuración y de incluir en las actividades del PGC, la estimación y planificación de las tareas asociadas a estos cambios. Las métricas son las siguientes:

- El control del avance del proyecto en la ejecución de cambios
- Errores de estimación de tiempo para los cambios
- Cumplimiento de la planificación del cambio

Se añaden a las métricas que fueron seleccionadas de la literatura, a partir de la vinculación de estos dos procesos una métrica adicional definida por Ailyn Febles en su tesis de doctorado [Febles, 2004]:

- Esfuerzo del personal involucrado en la resolución del cambio.

Esta métrica permitirá conocer y comparar la cantidad de cambios que tienen asignadas los miembros del equipo de proyecto. De esta manera es posible, al comparar a los diferentes miembros del equipo de proyecto y decidir sobre la base de estos elementos a quién se le asignará uno nuevo.



### 3.4.1.1 Métrica para el control del avance del proyecto en la ejecución de los cambios (AvPy)

Como ya fue planteado un plan detallado de configuración y de las tareas asociadas a los cambios esto ayuda a seguir y administrar el trabajo realizado. [Humphrey, 2000a] [Juran, 1994]

Según la literatura consultada unas de las principales mediciones para el seguimiento del proyecto son el cálculo del valor planificado (PV- Planed Value) y el valor devengado (EV- Earned Value). [Humphrey, 2000a] Estas mediciones proveen una forma de determinar la contribución de cada tarea al cronograma del proyecto. Durante la planificación se le llama valor planificado (PV- Planed Value) de una tarea o de una semana a lo que se planifica que esa tarea o semana contribuya al progreso del proyecto, y el valor de una tarea (o de una semana) será “devengado” solamente cuando esta tarea (o semana) se termine. No hay crédito parcial por tareas a medio concluir.

**AvPy** es una métrica definida por la Dr. Ailyn Febles, donde:

**P** es un conjunto finito de PV que se obtienen en uno de los proceso **PGC** y **GCC**

**Q** es un conjunto finito de EV obtenidos en el proceso **GCC**

**M**  $\in$  R, conjunto de los números reales

$\phi$  se calcula de la forma siguiente:

$$\phi : P \times Q \rightarrow M$$

$$\phi \left( \frac{p}{q} \right) = p/q \quad ; q \neq 0$$



### 3.4.1.2 Métrica para la determinación de errores en la estimación del tiempo (EEt)

Un método para medir la calidad de la planificación durante el desarrollo de un trabajo, o cuando se termine, específicamente para los cambios, es tomar las mediciones reales de tiempo y calcular el error entre las horas reales y las planificadas [Humphrey, 1995].

**EEt** es una métrica donde:

**P**, conjunto finito de *Tiempos Estimados* ( $te$ ) en uno de los procesos **PGC** y **GCC**

**Q** es un conjunto finito de *Tiempos Reales* ( $tr$ ) obtenidos en el proceso **GCC**

**M**,  $= \{m \in R : 0 \leq m \leq 100\}$ , donde R es el conjunto de los números reales.

$\phi$  Es una función que determina el error en la estimación del tiempo y se define de la forma siguiente:

$$\phi : P \times Q \rightarrow M$$

$$\phi(e, tr) = 100 \left( \frac{tr - te}{te} \right) \quad \text{donde } te \in P, tr \in Q$$

Será mejor la calidad de la planificación cuando el valor obtenido en **EEt** esté oscilando en valores cercanos a cero. Esta información podrá ser utilizada por los líderes de proyecto para futuras estimaciones de tareas similares. [Febles, 2004]

### 3.4.1.3 Métrica para medir el cumplimiento de lo planificado (CPI)

Esta métrica posibilita hacer un análisis del comportamiento del cumplimiento del plan del equipo o de un miembro en particular. Es similar a la anterior, la única diferencia radica en que esta es una razón y la otra un por ciento.

**CPI** es una métrica donde:



**P** conjunto finito de *Tiempos Estimados* ( $te$ ) en uno de los procesos **PGC** y **GCC**

**Q** es un conjunto finito de *Tiempos Reales* ( $tr$ ) obtenidos en el proceso **GCC**

**M** =  $\{m \in \mathbb{R}: 0 \leq m \leq 100\}$  donde  $\mathbb{R}$  es el conjunto de los números reales.

$\Phi$  Es una función que determina el error en la estimación del tiempo y se define de la forma siguiente:

$$\begin{aligned} \phi &: P \times Q \rightarrow M \\ \phi(e, tr) &= (tr - te) / te \end{aligned} \quad \text{donde } te \in P, tr \in Q$$

Esta métrica puede ser utilizada por los líderes de equipo para analizar el estado del proyecto y la vinculación de los miembros del equipo al desarrollo de las tareas. También puede ser gratificante para los desarrolladores ver el estado de su cumplimiento del plan. [Febles, 2004]

#### 3.4.1.4 Métrica para medir el esfuerzo de los miembros del equipo (EfP)

Permitir que el líder del equipo de proyecto tenga más elementos a la hora de asignar, reasignar tareas y de analizar el esfuerzo del personal asociado a los cambios en el proyecto, tiene que garantizarse en un modelo para la GC. Con este objetivo se definió la métrica para calcular el esfuerzo del personal.

**EfP** es una métrica donde:

**P** conjunto finito de tareas o tiempo planificado ( $tp$ ) en el proceso **PGC** o **GCC**

**Q** conjunto finito de miembros de un equipo ( $me$ ) u organización que participan en el proceso de **GCC**

$$M = \{(me, tp) | me \in Q, tp \in P\}$$



$\Phi$  se define de la forma siguiente:

$$\begin{aligned} \phi : Q \times P &\rightarrow M \\ \phi(e, tp) &= m \end{aligned} \quad \text{donde } m \in M$$

En esta métrica se calculan las tareas asignadas a cada miembro del equipo de proyecto. Se puede utilizar para uno o varios equipos de proyecto.

### 3.4.2 Métricas que vinculan la Identificación de los Elementos de Configuración y el Control de Versiones

La combinación de estos dos procesos no aparece en ninguno de los dos estándares tomados como referencia en la investigación, IEEE y CMM. Al no estar vinculados estos dos procesos la posibilidad de obtener métricas que arrojaran datos de la combinación de los elementos controlados en ambos procesos era imposible. El resultado de esta investigación de vincularlos en un sistema de métricas para la gestión de la configuración ha permitido controlar aspectos como los ECS asociados a las versiones de un producto, versiones generadas para cada ECS y estado en el que se encuentra en un momento dado un ECS [Febles, 2004].

Para garantizar el control de estos elementos fueron definidas por Ailyn Febles en su tesis de doctorado tres métricas que garantizan que durante todo el desarrollo del proyecto los involucrados se mantengan informados sobre el estado de los ECS. Estas métricas son las siguientes:

- ECS en cada versión del proyecto. (listados y cantidades)
- Estado de los ECS (libre o en uso) en un momento dado (listados y cantidades)
- Total de versiones de cada ECS registradas en la Línea Base.

Las tres métricas definidas para el sistema de GC son descritas en los próximos epígrafes.



### 3.4.2.1 Métrica: Elementos de Configuración del Software en cada versión de un proyecto. (ECpV)

Conocer los ECS que se encuentran en cada versión de un proyecto permite determinar en que medida se ha cumplido lo definido en el proceso de **IEC** y que ECS faltan por incluirse en el repositorio del proyecto. Listar estos elementos permite también verificar el estado de la línea base que debe coincidir con los ECS en el repositorio de una versión del proyecto.

**ECpV** es una métrica donde:

**P** es un conjunto finito de *ecs* establecidos en el proceso **IEC**

**Q** conjunto de  $v_{ij}$  de las *i* versiones del proyecto *j* generadas en el proceso **GVE**

$$M = \{(v_{ij}, ecs) \mid v_{ij} \in Q, ecs \in P\}$$

$\Phi$  se define de la forma siguiente:

$$\phi : Q \times P \rightarrow M$$

$$\phi \langle v_{ij}, ecs \rangle = m$$

donde  $m \in M$

Esta métrica resulta una información valiosa para el administrador de configuración y el líder del equipo de proyecto para conocer el estado actual de las componentes del producto en desarrollo. Se pueden obtener listados y totales con la información resultante de la métrica.

### 3.4.2.2 Métrica: Estado de los elementos de configuración (EEC)

Para el administrador de configuración y el líder del proyecto es imprescindible conocer en que estado están los ECS en un momento dado si *libre* o *en uso*. Obtener una métrica para conocer esta información



permite, por ejemplo, tener elementos a la hora de asignar una tarea asociada con un ECS o saber cuantos ECS están siendo modificados (*en uso*).

**EEC** es una métrica donde:

**P** es un conjunto finito de *ECS* establecidos en el proceso **IEC**

**Q** = {*en uso, libre*}, conjunto de estados en los que se encuentra un *ECS* en un momento dado y que varía en el proceso **GVE**

$$\mathbf{M} = \{(ecs, q); q \in Q, esc \in P\}$$

$\Phi$  se define de la forma siguiente:

$$\phi : P \times Q \rightarrow M$$

$$\phi \langle esc, q \rangle = m$$

$$\text{donde } q \in Q \wedge m \in M$$

### 3.4.2.3 Métrica: Versiones de cada Elementos de Configuración del Software registradas en la Línea Base (VLB)

La línea base del proyecto es uno de los artefactos fundamentales en un sistema de GC. Su estado varía cada un tiempo determinado en el proceso de planificación de la GC, por lo tanto es muy importante para los responsables del equipo conocer el estado de la línea base en un momento dado. También brinda información relevante para el progreso del proyecto conocer de cada ECS como ha sido registrado en la línea base. La métrica para obtener las versiones registradas en una línea base facilita el proceso de desarrollo de software. [Febles, 2004]

**VLB** es una métrica biproceso donde:



**P** es un conjunto finito de ECS establecidos en el proceso **IEC**

**Q** conjunto de  $v_{ij}$  de las  $i$  versiones de los ECS  $j$  de proyecto que se generan en el proceso **GVE**

$$M = \{(v_{ij}, ecs) \mid v_{ij} \in Q, ecs \in P\}$$

$\Phi$  se define de la forma siguiente:

$$\begin{aligned} \phi : Q \times P &\rightarrow M \\ \phi(v_{ij}, ecs) &= m \quad \text{donde } m \in M \end{aligned}$$

### 3.4.3 Métricas que vinculan la Gestión de los Cambios y el Control de Versiones

La combinación de estos dos procesos tampoco aparece en ninguno de los dos estándares tomados como referencia en la investigación, IEEE y CMM. El establecimiento por separado de estos procesos garantiza una mejor interpretación y ejecución de cada una de las actividades involucradas en ellos. Si embargo, es indiscutible la vinculación estrecha que tienen ambos procesos pues un cambio aprobado genera una versión nueva y una nueva versión es siempre generada por la existencia de un cambio.

Tener formalizados individualmente estos procesos pero manteniendo su vinculación en la implantación y ejecución del sistema de GC ha permitido que hayan sido definidas por la Dr. Ailyn Febles dos métricas importantes para el control de la configuración en los proyectos de desarrollo. Estas métricas no estaban reflejadas en la literatura por lo que resultan un aporte que se incluye en el sistema de métricas incluido en el modelo de referencia MConfig.pm y adaptadas para el modelo ModelConfig.UCI

Las métricas son las siguientes:

- Pedidos de cambio asociados a una versión del producto
- Versiones libres de defectos en una etapa



- Las dos métricas definidas son descritas a continuación.

### 3.4.3.1 Métrica: Pedidos de cambio asociadas a una versión de un producto (PcV)

Una versión de un producto que tenga muchos pedidos de cambios asociados indica, entre otro, que hubo una ineficiente definición de los requisitos funcionales o un mal diseño de la aplicación. En general puede significar insatisfacción de los usuarios. El líder del proyecto tiene que estar pendiente de las versiones de los productos y las peticiones de cambio de sus clientes. Que los procesos definidos brinden información que permita en cualquier momento conocer las versiones y los pedidos de cambio dará elementos a los directivos para tomar acciones correctivas o estratégicas en la etapa del proyecto en que se encuentren.

**PcV** es una métrica donde:

**P** es un conjunto *Pedidos de Cambio (pc)* recibidas en el proceso **GCC**

**Q** conjunto de versiones  $v_{ij}$  de un producto  $j$  generadas en el proceso **GVE**

$$M = \{(v_{ij}, pc); v_{ij} \in Q, pc \in P\}$$

$\Phi$  se define de la forma siguiente:

$$\begin{aligned} \phi : Q \times P &\rightarrow M \\ \phi \left( v_{ij}, pc \right) &= m \quad \text{donde } m \in M \end{aligned}$$

Esta métrica resulta una información valiosa para el Administrador de Configuración y el líder del Equipo de Proyecto para conocer el estado actual de las versiones de los productos implantados a los clientes. Se pueden obtener listados y totales con la información resultante de la métrica. [Febles, 2004]



### 3.4.3.2 Métrica: Versiones libres de defectos en una etapa (VLD)

Una versión estable no es necesariamente una versión libre de defectos. Una versión está libre de defectos, en un momento dado, cuando no hay ninguna petición de cambio o defecto detectado que no se encuentre resuelto. En la medida en que un producto de software se mantenga durante un mayor período de tiempo libre de defecto este producto es más robusto, hay mayor satisfacción de los clientes, etc. El Líder del Equipo debe estar midiendo constantemente la estabilidad de sus productos, la métrica VLD le facilita realizar esta medición.

**VLD** es una métrica donde:

**P** es un conjunto de *Problemas* ( $p$ ) recibidas en el proceso **GCC**

**Q** conjunto de versiones  $v_{ij}$  de un producto  $j$  generadas en el proceso **GVE**

$$\mathbf{M} = \{(v_{ij}) \mid v_{ij} \in Q\}$$

$\phi$  se define de la forma siguiente:

$$\begin{aligned} \phi : P \times Q &\rightarrow M \\ \phi(p, v_{ij}) &= m \quad \text{donde } m \in M \end{aligned}$$

También pueden obtenerse listados y totales con la información resultante de las métricas.

## 3.5 Sistema Informático de ModelConfig.UCI

La aplicación, de los procesos y las métricas de GC, se tornan extremadamente complejas si no se automatizan algunos de sus componentes. El modelo ModelConfig.UCI posee, además del Sistema de



Procesos y el Sistema de Métricas, un Sistema Informático. El sistema está compuesto por software propios y software del mercado internacional. [Febles, 2004]

El proceso de GC debe estar soportado por herramientas CASE, bien para apoyar alguna de las tareas del gestor, o bien integrando algunas o todas las tareas que se deben llevar a cabo.

El Sistema Informático (Fig. 3.1), del modelo propuesto, está compuesto por dos herramientas CASE y un sitio Web. Divide los cinco procesos del Sistema de Procesos en dos, un CASE para el control de versiones y otro para la ejecución del resto de los procesos. El sitio Web automatiza el Soporte de Software atendiendo principalmente las solicitudes de cambio [Febles, 2004].

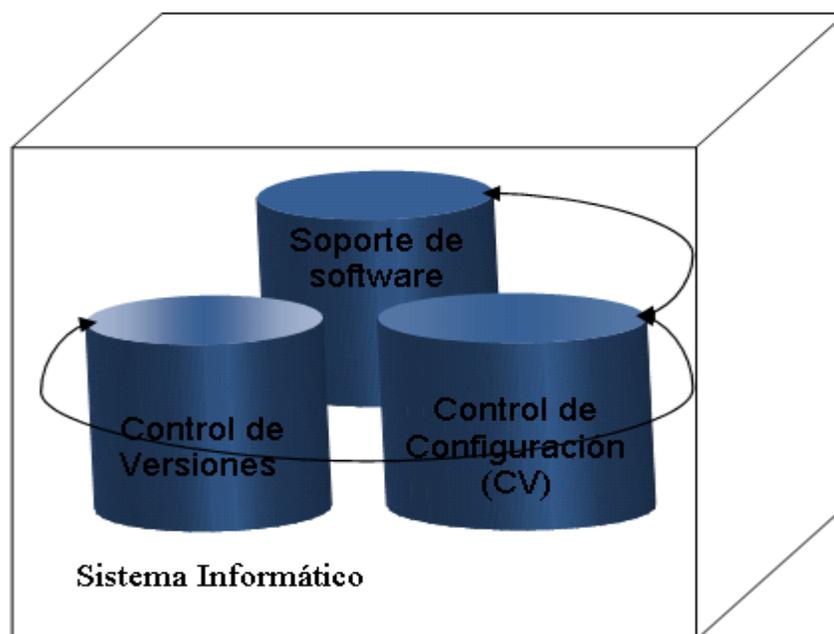


Figura 3.8 Sistema Informático de ModelConfig.UCI

El uso de las herramientas incluidas dentro del Sistema garantiza:

A los líderes de proyecto:



1. Prevenir el acceso no autorizado a los elementos
2. Coordinar, dar seguimiento y dirigir las actividades asociadas a los cambios

A los desarrolladores:

1. Conocer qué ha pasado con cada parte del código
2. Qué ha pasado con el resto de los componentes
3. Recordar cómo reproducir los errores
3. Conocer rápidamente las tareas asignadas para su resolución y almacenar el tiempo dedicado a estas.

A los directivos de la organización:

1. Posibilidad para prever la satisfacción del cliente
2. Posibilidad de revisar, actualizar versiones, etc. rápidamente
3. Conocimiento de todas las solicitudes de cambio

A los Clientes:

1. Mayor cercanía a la organización
2. Conocimiento, en tiempo, del estado en que están sus solicitudes
3. Mantenerse informado de los cambios en la organización, los nuevos productos, las nuevas versiones, etc.

### 3.5.1 Herramientas para la Gestión de Configuración

A continuación se efectúa en la presente sección un análisis de herramientas para la GC existentes en el mercado detallándose las prestaciones que brindan. Existen varias herramientas que integran, los procesos de GC [Technosoft, 2003]. Entre las más conocidas están, **IBM Rational ClearQuest**, **Rational ClearCase**, **Microsoft Visual SourceSafe (VSS)**, **Subversion (SVN)**, **Source Forge**. [Rational, 2000]



### 3.5.1.1 IBM Rational ClearQuest

IBM Rational ClearQuest permite efectuar seguimiento de defectos y cambios en toda la empresa [IBM, 2005a].

Entre las actividades que realiza Rational ClearQuest se encuentran:

- Seguimiento de cambios y defectos basado en actividades.
- Soporte para flujos de trabajo, que incluye notificaciones por correo electrónico y opciones de envío.
- Soporte para consultas con generación de múltiples informes y gráficos.
- Interfaz Web para acceder fácilmente desde cualquier navegador Web estándar.
- Integración con Rational ClearCase para conseguir una solución SCM completa.
- Integrado con los Entornos de Desarrollo (IDE) líderes en el sector, como **WebSphere Studio, Eclipse y Microsoft® .NET**

Una de las principales ventajas de Rational ClearQuest, como se puede apreciar en lo señalado anteriormente, es su integración con el resto de las herramientas de la **Suite de Rational** [Rational, 2003]. Sin duda una de las mayores dificultades de los productos Rational es su elevado costo. Rational ClearQuest cuesta \$1303.08 [IBM, 2005a]. Así como la alta generalidad con que cuenta en tareas como la generación de informes, que por el grado de flexibilidad y parametrización que posee, tiene una curva de aprendizaje alta, requiriendo de grandes esfuerzos para su asimilación.[Delgado, 2006]

### 3.5.1.2 Rational ClearCase

“IBM Rational ClearCase” es otra de las herramientas incluidas en la **Suite Rational 2003**. Ofrece una gestión de los activos de software para equipos de desarrollo de mediano y gran tamaño [IBM, 2005].

Entre las funcionalidades de Rational ClearCase están:

- Gestión del ciclo de vida y control de los activos de desarrollo de software



- Control integrado de versiones
- Gestión de Línea Base
- Integración con los líderes en el sector, como **WebSphere Studio, Eclipse y Microsoft® .NET**
- Integración con **Rational ClearQuest**.

Al igual que Rational ClearQuest, Rational ClearCase se integra con el resto de la Suite Rational y también comparte su elevado costo, alrededor de unos \$3319.38.

### 3.5.1.3 Microsoft Visual SourceSafe. VSS

**Microsoft Visual SourceSafe** (VSS) es una herramienta para el control de versiones que brinda soporte para puntos de restauración y capacidad de colaboración, que permite a los desarrolladores trabajar simultáneamente sobre una versión de un producto [MSDN, 2005]. MS Visual SourceSafe permite:

- Regresos (roll back) a versiones anteriores de un fichero.
- Ramificar, compartir, unir y administrar los entregables.
- Seguimiento de versiones de todo el proyecto.
- Seguimiento a código modular (un fichero que es usado o compartido por múltiples proyectos).
- Cuenta con las funcionalidades básicas de todo gestor de versiones (ultima versión, **“check In”**, **“check out”**)

El VSS, no presenta un costo elevado, en consecuencia con el soporte que brinda a las actividades de GC, pues sólo enmarca su trabajo en el control de versiones, por lo que para el desarrollo individual, podría resultar suficiente, mas al entrar a formar parte del trabajo en equipo, carece de un valor más allá del mero control de versiones ya que las funcionalidades para la coordinación del trabajo colaborativo no puede brindarla por si solo.

Otro elemento que no por último resulta menos importante es el carácter propietario de la herramienta, y su procedencia, al ser un producto de Microsoft, corporación estadounidense, no puede ser comercializado con Cuba. [Delgado, 2006]



#### 3.5.1.4 Subversion

A diferencia del VSS, **Subversion** es una herramienta libre “**open – source**” para el control de versiones. Y permite las funcionalidades que por lo general brindan los gestores de versiones más empleados:

Recuperar viejas versiones de los ficheros, ver su historial, etiquetar versiones, ramificar, unir, y efectuar retorno a versiones anteriores.

Es grandemente empleado por el mundo del desarrollo libre, siendo muy útil en este modelo de trabajo por lo general descentralizado y geográficamente disperso, en el cual se pondera mayormente la integración de los ficheros fuentes, en elementos más complejos, sin embargo para los modelos de trabajo en empresas, con responsables de proyectos, líderes de equipo, y trabajo coordinado, Subversion se queda al igual que VSS sin cubrir todas las necesidades de los equipos de trabajo. [Delgado, 2006]

#### 3.5.1.5 Source Forge

**Source Forge** es un ambiente de desarrollo colaborativo herramientas heterogéneas y procesos con un entorno integrado para la administración de proyectos, administración de cambios y capacidades de colaboración [SourceForge, 2005].

Entre las funcionalidades de Source Forge:

##### Herramientas de proyectos

- Seguimiento de problemas y cambios.
- Administración de tareas
- Sistema de liberaciones (release) de ficheros.

##### Colaboración

- Administración de documentos
- Fórum de discusión y noticias



- Lista de correos

### Interoperabilidad

- Microsoft Project
- Microsoft Office
- Sistema de Control de Versiones (Subversion, CVS, Rational ClearCase)

Como se ha visto en el análisis de las principales características y funcionalidades de estas herramientas para la GCS, si bien es cierto que todas estas dan soporte a muchas de las actividades incluidas en la GCS, dejan a un lado cosas muy importantes como la **identificación de los elementos de configuración** y la **administración de líneas base** y como habíamos planteado anteriormente muchas toman solo como GCS el control de versiones.

A partir de haber realizado un estudio de las herramientas existentes y tener un modelo definido para el control de configuración ModelConfig.UCI, atendiendo las desventajas de los software y el estado actual de los proyectos productivos de la UCI en el tema de GC, fue que se decidió utilizar una herramienta CASE(ConfigCASE 3.0) para automatizar los procesos y las métricas del modelo ModelConfig.UCI. [Febles, 2002]

## **3.5.2 ConfigCASE 3.0, herramienta CASE para el control de configuración**

### **3.5.2.1 Antecedentes**

Para desarrollar la herramienta CASE para el Control de Configuración, se partió de un CASE Corporativo (CASECorp) que en el año 2000 fue desarrollado por la Dr Febles. [Febles, 2002]

CASECorp automatizaba un flujo de trabajo que permitía el control de cambios a los elementos de configuración. Constituyó la primera aproximación a una herramienta para el control de configuración. [Febles, 2002]



Posteriormente con el objetivo de constituirse en un modelo de referencia para la Pequeña y Mediana Empresa (PyME) cubana de software, se propone un modelo Mconfig.pm. Como parte de esta propuesta fue desarrollado en el CRIS una herramienta CASE para el apoyo al proceso de la GCS en las (PyME), el cual fuera registrado en el Centro Nacional de Derecho de Autor (CENDA) en el año 2004 con el nombre: ConfigCASE 2.0.

ConfigCASE 2.0 pretende dar solución a problemas ubicados en dos direcciones: Control de Cambios y Asignación y seguimiento del trabajo en el proceso de control de configuración.

Revisando los objetivos de ConfigCASE 2.0 puede plantearse que el segundo fue implementado de manera satisfactoria. En cuanto al Control de la Configuración, ConfigCASE 2.0 se limita solamente al control de cambios a nivel de proyectos. Este hecho es consecuencia de la implementación parcial de los procesos planteados en [Febles, 2004].

De ahí que para dar solución a este problema es que [Delgado, 2006] se encargó de definir y construir la arquitectura de ConfigCASE 3.0 garantizando que soporte los procesos de Identificación y Control de cambios en la configuración definidos en MConfig.PM.

En la herramienta se automatizan los procesos definidos en MConfig.PM, modelo cuyo objetivo fue convertirse en el documento rector de la GCS en la InCuSoft, por lo que una vez concluida la construcción de la misma, se aplicó en las empresas. Las autoras proponen el uso de ConfigCASE 3.0 como herramienta a utilizar en los proyectos productivos de la UCI por todas las funcionalidades anteriormente expuesta [Delgado, 2006] [Febles, 2004]

### **3.5.2.2 Características generales de ConfigCASE 3.0**

Se definió ConfigCASE 3.0 como un software de apoyo al desarrollo del proceso de GCS y una plataforma para el control del flujo de trabajo. Se definieron siete módulos:

**Módulo General:** Contiene toda la información de los grupos de proyectos, desarrolladores, clientes, etc.



**Módulo de Metodologías:** Permite establecer las metodologías a seguir en la ejecución del proyecto, sus fases y etapas.

**Desarrollador:** Controla el proceso de resolución de los cambios en el proyecto. Permite además generar peticiones de cambios y defectos.

**Generador de Peticiones de Cambios:** Permite a cualquier especialista o líder de proyecto generar peticiones de cambios.

**Controlador de Peticiones de Cambios:** Permite al Líder de Proyecto asignar y evaluar las peticiones de cambio.

**Controlador de Órdenes de Trabajo:** Permite al Líder de Proyecto asignar y evaluar Órdenes de Trabajo durante todo el ciclo de vida del proyecto.

**Métricas:** Módulo que permite medir el proceso y determinar posibles mejoras en su ejecución.

### 3.5.3 Comparación de otros productos con el propuesto

Una herramienta muy utilizada en el mundo para control de proyectos es **Microsoft Project**. Resulta interesante incluirlo en esta comparación por la amplia difusión que tiene en el mundo para la planificación y seguimiento de proyectos.

La comparación en algunos elementos importantes para la GC, control de flujo de trabajo y mercado, entre algunas de las herramientas descritas en el epígrafe 3.6.1, Microsoft Project y ConfigCASE 3.0 se reflejan en la tabla 3.2. [Merant, 2003] [Project, 2003] [Rational, 2003] [Continuus, 2003] [Rational, 2000]



Tabla 3.2 Comparación entre herramientas CASE

	Project	Rational Clear CASE	ConfigCASE 3.0
Control de configuración	no	si	si
Alcance	Proyectos simples y complejos	Proyectos simples y complejos	Proyectos simples y complejos
Facilidad de Uso	Flexible, eficaz y fácil de usar	No es cómodo y fácil de utilizar	Fácil y cómodo de utilizar
¿Control de cambios a?	Cambios en la planificación	Control de cambios en general	Control de cambios en general
Control de flujo de trabajo	si	no	si
Costo en el mercado	\$588	\$ 5 179 (1 PC)	-

### 3.5.4 Consideraciones para el desarrollo de la herramienta

El proceso de desarrollo se realizó siguiendo las especificaciones de **Rational Unified Process** (RUP) [Jacobson, 2000] [Rumbaugh, 1999] y para la documentación generada se utilizó UML (Unified Modeling Language) que se ha convertido en un estándar internacional generalizado para el desarrollo de aplicaciones con la tecnología orientada a objetos.

### 3.5.5 Herramienta CASE para el Control de Versiones



A partir de que se incluye en el Sistema Informático de MConfig.pm y adaptada a ModelConfig.UCI la herramienta CASE ConfigCASE 3.0 resultaría necesario incluir una herramienta para el control de versiones. Una herramienta CASE para el control de versiones debe permitir [Butler, 2001] [Febles, 2004]:

- Identificación de versiones y entregas.
- Gestión de almacenamiento.
- Registro de historial de cambios.
- Desarrollo independiente.

Existen varias Herramientas CASE en el mercado para el control de versiones, entre las más difundidas, utilizadas y con grandes prestaciones para la ejecución de este proceso es Subversion (SVN), es software libre bajo una licencia de tipo Apache/BSD. Se establece SVN como herramienta CASE para el control de versiones del Sistema Informático de ModelConfig.UCI.

Fueron consideradas las siguientes razones para seleccionar SVN:

Los archivos versionados no tienen cada uno un número de revisión independiente.

Todo el repositorio tiene un único número de versión que identifica un estado común de todos los archivos del repositorio en cierto punto del tiempo.

Se sigue la historia de los archivos y directorios a través de copias y renombrados.

Las modificaciones (incluyendo cambios a varios archivos) son atómicas.

La creación de ramas y etiquetas es una operación más eficiente; Tiene costo de complejidad constante ( $O(1)$ ) y no lineal ( $O(n)$ ) como en **Concurrent Versions System** (CVS).

Se envían sólo las diferencias en ambas direcciones (en CVS siempre se envían al servidor archivos completos).



Puede ser servido, mediante Apache, sobre **WebDAV/DeltaV**. Esto permite que clientes WebDAV utilicen Subversion en forma transparente.

Maneja eficientemente archivos binarios (a diferencia de CVS que los trata internamente como si fueran de texto).

Permite selectivamente el bloqueo de archivos. Se usa en archivos binarios que, al no poder fusionarse fácilmente, conviene que no sean editados por más de una persona a la vez.

Cuando se usa integrado a Apache permite utilizar todas las opciones que este servidor provee a la hora de autenticar archivos (SQL, LDAP, PAM, etc) [autores, 2002].

### **3.5.6 Sitio Web para el Soporte de Software**

Adicionalmente a las dos herramientas descritas en los epígrafes anteriores, con el objetivo de lograr avances en el servicio de Soporte de Software en los proyectos productivos y darle un valor agregado al Sistema Informático del modelo propuesto, se decidió utilizar para el Soporte del Software, SoSoft, el sitio Web parametrizable desarrollado en el MConfig.pm. La herramienta brinda una forma cómoda para almacenar y manejar información relacionada con esta actividad en la organización. [Febles, 2002]

SoSoft consta de dos módulos, el Módulo Cliente y el Módulo Administrador.

El Módulo Administración se ocupa del trabajo del Grupo de Soporte de Software. En él se brindan servicios como: actualización de la información a publicar en el Sitio de Soporte de Software, respuesta a las solicitudes de los clientes, obtención de la información del Sitio de Soporte de Software, envío de correo electrónico sobre una nueva versión, actualización de los usuarios del sistema, actualización de la información de la organización y establecer los parámetros en el Sitio.

El Módulo Cliente es un sitio para el Soporte de Software de la organización a publicar en Internet o en la Intranet. Es la cara de la herramienta que se le muestra al cliente de la organización. Brindar los siguientes servicios como: Soporte por productos, Soporte Personalizado, Soporte por cliente, Preguntas



más frecuentes (FAQs), Búsquedas Avanzadas, Inquietudes, Foros de Discusión, Contactos, solicitud de productos nuevos o existentes y la Solicitud de los cambios. [Febles, 2002]

El sitio, al implantarse en cada proyecto, puede configurarse para que se conecte a la base de datos de ConfigCASE 3.0 lográndose que el proceso de Gestión de Cambios quede automatizado desde el momento en que el usuario solicita el cambio. De esta manera, el tiempo entre que el cambio es detectado y que llega la solicitud a la organización se acorta, y de igual manera el usuario se mantiene actualizado sobre el estado en que se encuentran sus solicitudes.



## Conclusiones parciales

El modelo ModelConfig.UCI está formado por tres sistemas, un sistema de procesos, un sistema de métricas y un sistema informático.

El sistema de procesos está formado por cinco procesos para la GC. Su tratamiento en conjunto dentro de un Sistema de GC es totalmente nuevo dentro de esta área del conocimiento y con aplicaciones prácticas en la UCI se comprobará que la acción simultánea de los cinco procesos produce una sinergia beneficiosa para gestionar los ECS.

El Sistema de Métricas en métricas ha permitido una mejor asimilación de esta por parte de desarrolladores y directivos y establecer una relación clara con cada uno de los procesos definidos en el sistema de Procesos.

El sistema informático propuesto está formado por tres herramientas: una para el control de versiones, otra para el resto de los procesos de GC y un sitio Web para el Soporte de Software.

El acelerado desarrollo de la industria del software a nivel mundial ha llevado a que el proceso de desarrollo sea apoyado por un grupo de herramientas, que van a automatizar los procesos definidos dentro del proceso productivo. Dichas herramientas reciben el nombre de herramientas CASE. En la actualidad a nivel internacional existen diversas herramientas que automatizan algunas de las actividades que componen la GCS muchas de estas presentan un costo de adquisición y asimilación extremadamente elevado.

El ConfigCASE 3.0 reduce el esfuerzo requerido para realizar las actividades asociadas a la GC además brinda los resultados de la evaluación de las métricas definidas en el modelo una forma clara y concisa al incluir el tratamiento gráfico de la información.

El sitio Web permite acercar al cliente al proyecto y en particular lo hace formar parte del proceso de GC permitiéndole solicitar directamente los cambios y conocer su estado.



El modelo de referencia de la GC para los proyectos productivos constituye una forma de suplir la carencia de tales herramientas en la UCI y su integración con otros modelos y estándares puede contribuir a elevar los niveles de madurez de los mismos.



## Conclusiones

Se obtuvo un modelo de referencia para la GC **ModelConfig.UCI** formado por tres sistemas, un *Sistema de Procesos*, un *Sistema de Métricas* y un *Sistema Informático* y por el resto de los elementos que le son inherentes: objetivos, entradas y salidas.

El Modelo de Referencia para la GC, **ModelConfig.UCI** contribuye a suplir la carencia de modelos de este tipo, que existen actualmente en la UCI, que integrado con otros modelos y estándares permitirá elevar los niveles de madurez de los proyectos productivos en la universidad. El modelo permite tomar acciones correctivas, garantiza mejor comunicación entre los involucrados e implica mayor calidad en la ejecución de los procesos y por tanto mejor calidad del producto final.

Con la aplicación práctica del modelo se logrará establecer disciplina de trabajo en las tareas asociadas a los cambios, las versiones y los ECS; medir atributos de calidad mejorando sus resultados durante el proceso de implantación al poder tomar medidas correctivas a tiempo y almacenar los resultados para la toma de decisión en futuras tareas o proyectos. Haber propuesto el modelo de forma automatizada permitirá que los miembros de los proyectos no rechacen la disciplina de trabajo establecida y que se obtenga mayor eficiencia y calidad en la ejecución de los procesos y el cálculo de las métricas.

Implantando el modelo **ModelConfig.UCI** es posible controlar y asegurar la calidad de la producción de software y mantener una historia de los desarrollos realizados. Las métricas del modelo garantiza la mejora incremental de los procesos y su automatización permite que la transmisión de la información sea rápida y fiable haciendo eficiente el flujo de trabajo.

Para el **Sistema de Procesos** de ModelConfig.UCI se seleccionaron cinco procesos para la GC. Su tratamiento en conjunto dentro de un Sistema de GC es totalmente nuevo en esta área del conocimiento. Además produce una sinergia beneficiosa para gestionar los ECS.



El **Sistema de Procesos** se diseñó utilizando la definición de procedimientos y diagramas basados en estándares internacionales facilitando su implantación, comprensión y ejecución de manera gradual e integrada.

La división del **Sistema de Métricas** en dos grupos: métricas biprocesos y métricas uniprocesos y su definición formal, facilitó el desarrollo de la investigación y representa una modesta contribución a la mejor asimilación de las métricas por parte de desarrolladores y líderes. Permitió además establecer una relación clara de las métricas con cada uno de los procesos definidos en el Sistema de Procesos.

El **Sistema Informático** de ModelConfig.UCI, integrado por dos herramientas CASE: una para el control de versiones y otra para el resto de los procesos de GC así como un sitio Web para el Soporte de Software, constituye un componente básico y útil para lograr implantar un Modelo de GC en los proyectos productivos de la UCI.

Se comprobó que:

**ConfigCASE 3.0**, reduce el esfuerzo requerido para realizar las actividades asociadas a la GC, además brinda los resultados de la evaluación de las métricas definidas en el modelo de forma clara y concisa al incluir el tratamiento gráfico de la información.

**Subversion**, constituye una poderosa herramienta para el control de versiones, y **SoSoft**, acerca al cliente al proyecto y en particular lo hace formar parte del proceso de GC permitiéndole solicitar directamente los cambios y conocer su estado.



## **Recomendaciones**

Aplicar el modelo, sistemas y herramientas para la GC en todos los proyectos productivos de la UCI y evaluar sus resultados. Para esto será necesario escoger una muestra inicial, valorar los resultados, realizar las mejoras y luego en correspondencia con las mismas aplicarlo en el resto de los proyectos.

Enriquecer el Sistema de Métricas con la inclusión de propuestas específicas orientadas al tamaño utilizando puntos de función y casos de uso y métricas orientadas al cliente.

Diseñar e implementar la herramienta ConfigCase 3.0 para los restantes procesos definidos en ModelConfig.UCI y así mejorar las prestaciones del Sistema Informático del modelo.

Definir e implementar un sistema de indicadores que permita calcular el impacto del cambio en un proyecto.

El desarrollo de este modelo debe ser planificado y estandarizado para aplicarlo a todos los proyectos que se desarrollen con independencia de su tamaño, de forma que pueda ser interiorizado por la UCI, lo cual redundará en mayor calidad del producto.



## Referencias Bibliográficas

[Álvarez, 2000] Álvarez S. Febles A. Fernández H. Aplicación del modelo CMM a la empresa Segurmática, Ingeniería Industrial, ISPJAE 2000

[Álvarez, 2003] Álvarez S, Un modelo de calidad para una empresa de software, Evento Calidad 2003 Convención Informática 2003, C. Habana (CU), 2003.

[AMCIS, 2001] AMCIS, BANCOMEX, AMITI,. Importancia del desarrollo de tecnologías de información y comunicaciones (tic) en México, Soluciones Empresariales, 2001. Disponible en: <http://www.claveempresarial.com/soluciones/notas/nota010827a.htm>

[Antonio, 2001] de Antonio Angélica, Gestión de Configuración, Chile, SPIN, 2001

[Appleton, 2000] Appleton Brad. SCM definitions, 2000, Disponible en: <http://www.enteract.com/~bradapp/scm-defs.html>

[autores, 2002]. Wikipedia.La enciclopedia libre. Recuperado el 30 de mayo de 2007, Disponible en: <http://www.es.wikipedia.org/>

[Baeza, 1995] Baeza Yates, R. A. et. al. Computing in Chile: ¿The Jaguar of the Pacific Rim?, Communications of the ACM. Vol. 38, No 9, 1995.

[Bedini, 1995] Bedini Alejandro G. "Calidad de software", Novática,1995

Brito, I. (2007, 5 14). ¿Cómo se realiza la Gestión de Configuración es su proyecto? (Y. L. Perdomo, Interviewer

[Butler, 2001] Butler Tresa, Faith Turner, Verla Standley, Elaine Sullivan, Software Configuration Management: A discipline with added value, Crosstalk Magazine, 2001 Jul, Vol. 7.



[Cadie, 2001] CADIE, LA industria de software como industria global, 2001

[Camou, 2002] Camou Elena. Software Development: The Venezuelan competitive project, cavecom-e, 2002 p. 7-14

[CaproSOFT, 2002] CaproSOFT, Programa de apoyo a la competitividad del sector de software, 2002. Disponible en: <http://www.Caprosoft.org>

[CITMA, 2002] Ministerio de Ciencia, Tecnología y Medio Ambiente., Algunas experiencias internacionales en el desarrollo y explotación de software, 2002 dic.

[cmBook, 1998] Brown Norm; W. Evans Michael; Gathmann-Hobbs Jeannine. Little Book of Configuration Manager. Computer & Concepts Associates, 1998.

[CMMI 2005] CMMI (2005). The Capability Maturity Model Integrated. USA, Carnegie Mellon Software Engineering Institute.

[Continuus, 2001] Continuus, Gestión de Cambios y Configuración en el desarrollo de Software, 2001. Disponible en: <http://www.continuus.com>

[Delgado, R. febrero, 2007]. Situación Actual de la Gestión de Configuración en los proyectos productivos de la UCI. (Y. M. Yeniset León, Interviewer)

[Febles, 2000a] Febles Ailyn, "Case Corporativo para el proceso de control de cambios". [Tesis de Maestría]. Ciudad de la Habana (CU), CUJAE, 2000.

[Febles, 2000b] Febles A., Álvarez S. Modelo de calidad CMM y su aplicación en las empresas, Memorias del primer encuentro internacional de Computación aplicada, Guadalajara, México, 2000

[Febles, 2000c] Febles A. Álvarez S. Fernández H. "CASE Corporativo para la creación de la línea base de una empresa de Software", Revista Ingeniería Industrial, CUJAE, 2000



[Febles, 2002] Febles Ailyn, Un sitio de soporte de software, acercando la empresa al cliente, Memorias de la Convención Informática 2002, La Habana (CU), 2002 feb.

[Febles, 2003] Febles A., “Guía práctica del modelo de referencia MConfig.pm”, Ingeniería Industrial, cujae, 2003

[Febles, 2004] Febles Ailyn, MConfig.PM, Modelo de referencia para la Gestión de Configuración en la pequeña y mediana empresa de software La Habana, Cuba, CUJAE, 2004

[Fernández, 1999] Fernández Sanz Luis, Alarcón Rodríguez Miren I., Necesidades de medición en la gestión y el aseguramiento de calidad del software, Universidad Europea de Madrid, Universidad Autónoma de Madrid, 1999

[Franco, 2006] Franco José Ángel, “Entorno Unificado para la Gestión de Configuración de Software”, La Habana, Cuba, CUJAE, 2006

[Freeman, 1990] Freeman, D.P., Weimberg, G.M. *Handbook of walkthroughs, Inspections and Technical Reviews*, 3rd Edition, Dorset House, 1990.

[García, 2000] García Lourdes, Metodología para la evaluación de la calidad del análisis y diseño orientado a objeto usando ADOOSI [Tesis de doctorado], C. Habana, , CEIS, ISPJAE, 2000

[Gartner 2001] Gartner, Describing the Capability Maturity Model, Gartner, Inc. 2001

[Gervás, 2002], Gervás Pablo, Estándares y Gestión de Configuración, Capítulo 1, UCM, 2002, p4.

[Ginsberg, 1995]. Mark Ginsberg and Lauren Quinn, “Process Tailoring and the Software Capability Maturity Model,” Software Engineering Institute, CMU/SEI-94-TR-024, November 1995.

[González, 2003] González P. Ignacio, Discurso pronunciado en la inauguración de la Convención Informática 2003, Cuba, 2003



[GTI, 2002], Reporte de Trabajo No. 2. Comisión de Productos y Servicios-Fuerza de Tareas de la Industria de Software, GTI, MIC, Rev00, septiembre 2002.

Guido, E. F. (2007). GUÍA PARA DISEÑAR Y PROCESAR ENCUESTAS EN ORGANISMOS PÚBLICOS. Buenos Aires: Unidad de Información y Comunicación.

[Haque, 1995] Haque Sohail. Introducing Process into Configuration Management, IBM's AIXPERT Magazine, 1995 Nov.

[Hernández, 1996] Hernández A. "Automatización del Diseño", [Tesis de Maestría], CEIS, ISPJAE, enero 1996

[Humphrey, 1989] Humphrey Watts S.. "Managing the Software Process". Addison- Wesley, 1989

[IBM 2005a] Rational ClearCase, International Business Machine, Disponible en: <http://www.ibm.com/software/awdtools/clearcase/>, diciembre de 2005

[IEEE, 1987] IEEE, IEEE Guide to Software Configuration Management, American National Standards Institute, 1987, Std. 1042-1987

[IEEE, 1990] IEEE, IEEE Standard for Software Configuration Management Plans, American National Standards Institute, 1990, Std. 828-1990

[IEEE 1998] "IEEE Standard for Software Quality Assurance Plans". Std. 730-1998, IEEE Computer Society, 1998.

[ISO, 1995] ISO. ISO 10007 Quality management – Guidelines for configuration management, ISO, 1995 abr. 15.

[Jacobson, 2000], I. El Proceso Unificado de Desarrollo de Software, Addison Wesley Longman Inc, 2000.



[Jalote, 2002a] Jalote Pankaj. Software Process Improvement, Primer Taller Internacional de Calidad en desarrollo de software, México, 2002 may.

[Jorge, 2003] Jorge Claudio, Gestión de Configuración de productos software en etapa de desarrollo, julio, 2003.

[Lage, 2000] Lage Carlos, Discurso pronunciado en la Inauguración de la Convención Informática 2000, Cuba, 2000.

[Marín, 2002] David Marín. Control de Configuración, 2002, Disponible en: [http://fact.aspl.es/doc/doc\\_html/gcs\\_noe4.html](http://fact.aspl.es/doc/doc_html/gcs_noe4.html)

[Merant, 2003] MERANT, Ofertas de precio, 2003, Disponible en: <http://www.cstores.uiuc.edu/center/prices/specials/0805CSSW.html>

[Michelena, 2001] Michelena Esther, La Gestión de Procesos en el Sistema de Gestión de Calidad ISO 9000:2000, Revista Ingeniería Industrial, 2001.

[Ministerio de Economía y Producción República Argentina, 2004]. Proargentina. Recuperado el 3 de 4 de 2007. Disponible en: <http://www.proargentina.gov.ar>

[Moreno, 2003] Moreno Boris, Conferencia magistral, Evento de Calidad, Convención Informática 2003, Cuba, 2003

[MSDN 2005] Microsoft Visual SourceSafe Roadmap, Microsoft Development Network, Microsoft Corp., 2005

[Oktaba, 2001a] Oktaba Hanna, Claudia Alquicia, Modelo gráfico de la administración de Configuraciones del SW-CMM nivel 2, UNAM, México, 2001



[Oktaba, 2001b] Oktaba Hanna, Modelo gráfico de la administración de Requerimientos del SW-CMM nivel 2, UNAM, México, 2001

[Oktaba, 2003] Oktaba Hanna, El modelo de madurez de capacidades mexicano, Conferencia pronunciada en el evento Calidad 2003 en la Convención Informática 2003, Cuba, 2003

[Paulk, 1993] Paulk M.C., et al, Capability Maturity Model for Software, Version 1.1, Carnegie Mellon University, SEI-93-TR-024. 1993

[Paulk, 1999a] Paulk M. C., Weber C.V., et. al., The Capability Maturity Model, Guidelines for Improving the Software Process, p. 180–191, CMU, SEI, 1999

[Paulk, 1998] Paulk Mark. Using the CMM in small organizations, SEI, CMU, 1998 p.4

[Pressman, R. S], Ingeniería del Software. Un enfoque práctico. Mc Graw Hill.

[Preesman, 1997] Pressman Roger S., Ingeniería de Software, un enfoque práctico, ed. 3, Mc GrawHill, 1997

[Pressman 1998] Pressman, R. Ingeniería de software. Un enfoque práctico. Madrid, Mc Graw-Hill Interamericana de España S.A., 1998

[Preesman, 2000], Pressman Roger S., Ingeniería de Software, un enfoque práctico, ed. 4, Mc GrawHill Iberoamericana 2000 [Project, 2003] Project, Ofertas de precio, 2003, Disponible en: <http://simplestshop.com/>

[Project, 2003] Project, Ofertas de precio, 2003, Disponible en: <http://simplestshop.com/software/product/491286/B000066EUO/0/1/Microsoft%2520Project%25202002.html>



[Rational, 2000] Rational Software Corporation, Importancia de la Administración de la Configuración de software, 2000. Disponible en: <http://www.rational.com>

[Rational, 2003] Rational, Ofertas de precio, 2003, Disponible en: [http://www.indudata.com/lprecios\\_productos.htm#1](http://www.indudata.com/lprecios_productos.htm#1)

[Rational 2004] IBM RedBooks. Software Configuration Management A Clear Case for IBM Rational ClearCase and ClearQuest UCM. Publicado en diciembre del 2004. ISBN: 0738491594.

[Reo, 2002] Reo Davis A., La gallina, el cerdo y el modelo CMM, América XXI, 2002 sep.-4, Disponible en: <http://www.americaxxi.cl/modules.htm>

[Ruiz, A, 2007], La Universidad de la Ciencias Informáticas y la Industria del Software. *La Universidad de la Ciencias Informáticas y la Industria del Software*, (pp. 1-24). Ciudad Habana.

[Rumbaugh, 1999] Rumbaugh, J., I. Jacobson and G. Booch, The Unified Modeling Language, Reference Manual, Addison-Weslwy, 1999

[SEI, 2000] SEI, Software Engineering Measurement and Analysis Team, Process Maturity Profile of the software community 2000 mid-year update, 2000

[Sorensen, 1999] Sorensen Reed. ¿CCB-An Acronym for “Chocolate Chip Brownies”?, Crosstalk Magazine, 1999 mar. Disponible en: <http://www.stsc.hill.af.mil/crosstalk/1999/03/sorensen.asp>

[SourceForge 2005] VA Software Corporation. Disponible en : <http://www.sourceforge.org>, noviembre del 2005

[Takahashi, 2001] Takahashi Tadao, et. al. Sociedad de la Información en Brasil, Libro Verde Brasilia, Ministerio de Ciencia y Tecnología, ed. Sociedad de la Información (SocInfo), Octubre, 2001



[Technosoft, 2003] Technosoft, Software Configuration Manager, 2003, Disponible en: <http://www.technosoft.com/OurServices/CM/config.htm>

[Torres, 2001] Torres Elizabeth, Proceso de desarrollo de software en Chile, INGES PUCC, Julio 2001,

[Villamil, 1999], Villamil Gustavo. ISO 9000 aplicada al software. Tercera parte, Guías de la computación, 1999, p3

[Wiegers, 1998] Wiegers, K. Molding the CMM to your organization. Software Development. May 1998, v6, n5, p49

[Zhang, 2001] Zhang Moxie J. CMM-SW and ISO 9001. The ending of the Hero Era, 2001, Disponible en: <http://www.eMoxie.com>



## ANEXOS

### Anexo 1

Datos de la Industria de software en algunos países (2000)

Indicadores	Brasil	Chile	Costa Rica	India	Irlanda	Israel	Uruguay
Ventas (millones de USD)	8,038	125	---	5,700	8,800	3,700	180
Exportaciones (millones de USD)	40	15	50	4,000	7,369	2,600	60
Exportaciones (% Ventas)	0.50	12.00	---	70.18	83.74	70.27	33.33
No. Empleados (miles)	---	---	3.75	410.00	30.00	14.50	2.75
Productividad (miles USD/empleados)	---	---	---	13.90	293.33	255.17	65.45



## Anexo 2

### Listado de los proyectos productivos encuestados por facultades

Facultad	Proyectos Productivos
1	Akademos
1	Empresa Correos de Cuba – Correos Venezuela (ECC – IPOSTEL)
2	Identidad
2	PROCYON
2	Operación Verdad
2	Centro de Tratamiento de Atención de la Información de Seguridad Ciudadana (CTAISC)
3	Inventario
3	Oficina Nacional de Estadística (ONE)
3	Mixta Cuba - Venezuela
4	Sistema de Gestión Penitenciaria (SIGEP)
4	Ministerio de las Fuerzas Armadas Revolucionarias (MINFAR)
5	Quirúrgico
5	SCADA de PDVSA
6	Ensayos Clínicos



6	Plataforma para la Predicción de Actividades Biológicas (GRaph-TOol)
6	Software para la Simulación de Sistemas Biológicos (BioSyS)
7	Grupo de Procesamiento de Imágenes (GPI)
7	Atención Primaria de la Salud (APS)
8	Centro Nacional de Tecnologías de Información (CNTI)
8	Cuerpo de investigaciones Científicas, Penales y Criminalísticas (CICPC)
9	A Jugar
9	Operación Verdad
9	Control y Gestión de Indicadores
10	Distribución de Linux (Nova)
10	Archivo General de la República de Venezuela
10	Intranet de PDVSA

### **Anexo 3**

**Encuesta sobre la disciplina de Gestión de Configuración a proyectos productivos de las facultades de la Universidad de las Ciencias Informáticas.**



Nombre del proyecto: \_\_\_\_\_

Facultad: \_\_\_\_\_

Rol del encuestado: \_\_\_\_\_

Año de creación: \_\_\_\_\_

Tipo de proyecto: \_\_\_\_\_

Tamaño del equipo de desarrollo: \_\_\_\_\_

1. ¿Se dedican solo a la producción de software? Si\_\_\_ No\_\_\_

a. ¿Qué otras actividades realizan?

\_\_\_\_\_  
\_\_\_\_\_

2. ¿Cuántos productos han fabricados? \_\_\_\_\_

3. Su producto está destinado al beneficio de(Especifique a que proyecto u organización le brinda servicio):

a. \_\_\_ La UCI ( )

b. \_\_\_ Organizaciones cubanas que necesitan software para desarrollarse ( )

c. \_\_\_ Organizaciones extranjeras que necesitan software para desarrollarse ( )

d. \_\_\_ Otros( )



4. La calidad de los productos que fabrican es: E\_\_\_ B\_\_\_ R\_\_\_ M\_\_\_

5. ¿Qué usted entiende por calidad?

---

---

---

6. ¿Tiene en su proyecto un equipo de aseguramiento de la calidad del software?

Si\_\_\_ No\_\_\_ No sé\_\_\_

Si la respuesta es Si

a. ¿Tiene un personal dedicado específicamente a la Gestión de Configuración (GC)?:

Si\_\_\_ No\_\_\_ No sé\_\_\_

i. ¿Qué usted entiende por GC?

---

---

ii. ¿Tiene en su proyecto el rol de Administrador de Configuración?:

Si\_\_\_ No\_\_\_ No sé\_\_\_

Si la respuesta es Si



Diga tres tareas fundamentales de ese rol.

1. \_\_\_\_\_
2. \_\_\_\_\_
3. \_\_\_\_\_

b. ¿Aplica algún Plan de GC en su proyecto?

Si\_\_\_ No\_\_\_ No sé\_\_\_

c. ¿Cómo identifica y gestiona su proyecto las versiones existentes de su programa y su documentación?:

---

---

i. Seleccione las herramientas que utiliza.

\_\_\_ Visual Source Safe

\_\_\_ Subversion

\_\_\_ Clear Case

\_\_\_ Otras

d. ¿Cómo controla su proyecto los cambios antes y después de que el software sea distribuido al cliente?:

---



7. Con la disciplina de GC se logra:

- a.  Mayor satisfacción de los clientes  
 No influye en la satisfacción de los clientes  
 Menor satisfacción de los clientes
  
- b.  Mayor organización y planificación del trabajo  
 No influye en la organización y planificación del trabajo  
 Menor organización y planificación del trabajo
  
- c.  Aumenta el tiempo que los estudiantes dedican a la fabricación de software  
 No influye en el tiempo que los estudiantes dedican a la fabricación de software  
 Disminuye el tiempo que los estudiantes dedican a la fabricación de software
  
- d.  Aumenta la cantidad de clientes  
 No influye en la cantidad de clientes  
 Disminuye la cantidad de clientes

8. ¿Es necesario la GC?



Si \_\_\_ No \_\_\_

a. Mencione alguna de sus ventajas.

---

---

9. Conoce si su facultad tiene definido algo respecto a la GC en sus proyectos productivos (Fundamente su respuesta).

Si \_\_\_ No \_\_\_ No sé \_\_\_

---

---

Son de gran interés las respuestas que nos ha proporcionado.

Muchas gracias por dejarnos robar un poco de su tiempo.



## Anexo 4

Entrevista realizada a cuatro de los principales proyectos productivos de las universidad.

Facultad	Nombre del Proyecto Productivo	Nombre del entrevistado	Rol que desempeña
1	Identidad	Irina Brito	Líder de Proyecto
3	Registro y Notaría (R&N)	José Luis León	Líder de Proyecto
4	Sistema de Gestión Penitenciaria (SIGEP)	Ana Marys García	Gestor de Cambio
5	SCADA de PDVSA	Amado Espinosa	Líder de Proyecto
Principales preguntas realizadas			
1	¿Aplican en su proyecto la disciplina de Gestión de Configuración?		
2	¿Tienen definido algún modelo de Gestión de Configuración por el cual regirse?		
3	Explique en que consiste dicho modelo		

## Anexo 5

### Ejemplos de Elementos de Configuración de Software

#### A. Datos de un elemento de configuración de software

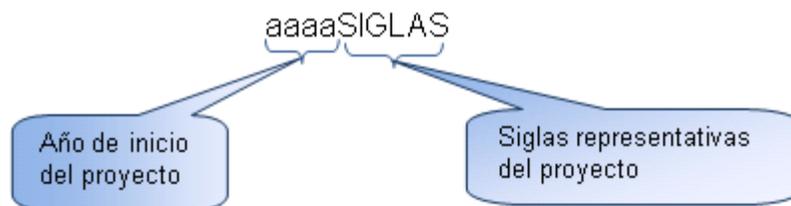
1. Número o código del Elemento de Configuración del Software.
2. Nombre del ECS
3. Descripción del ECS
4. Autor/es del ECS
5. Fecha de creación
6. Identificación del proyecto al que pertenece el Elemento de Configuración.



7. Identificación de la línea base a la que pertenece.
8. Identificación de la fase y subfase en la que se creó.
9. Tipo de Elemento de Configuración (documento, programa, elemento físico, etc.)
10. Localización

B. Esquema para la ubicación en el repositorio

- Existirá en el repositorio una carpeta por cada proyecto de la organización
- El nombre de las carpetas tienen la siguiente sintaxis:



- Dentro tendrá subcarpetas para cada fase y para actividades típicas que sea necesario diferenciar los documentos. Se nombrarán con la misma sintaxis que la carpeta del proyecto, las siglas serán las representativas de la fase o la actividad típica. Por ejemplo como el de la Fig. A 5.1

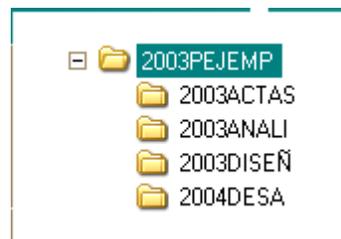
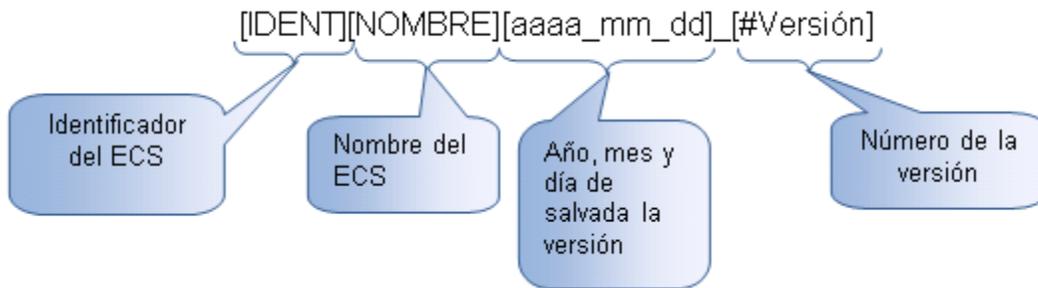


Figura A 5.1

C. Esquema para nombrar un ECS



Para nombrar los Elementos de Configuración se utilizará la siguiente sintaxis:



El identificador del ECS puede ser:

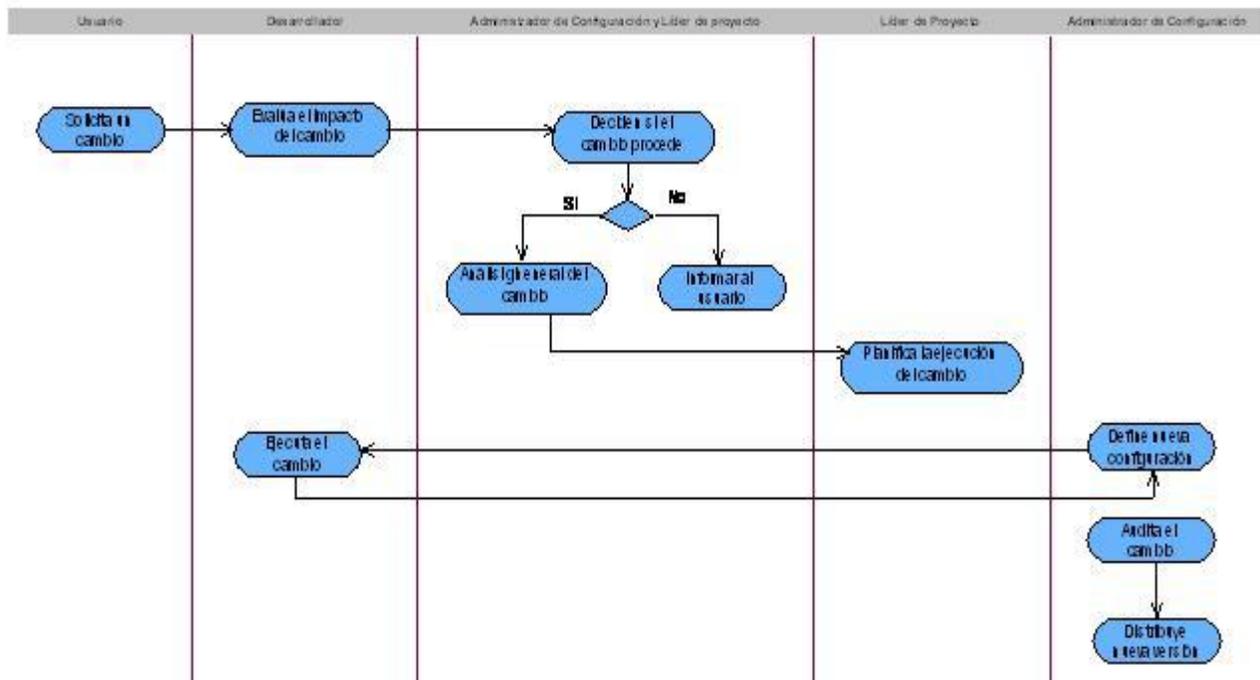
- BD → Base de datos
- T → Textos
- Z → Ficheros compactados
- F → Fuentes
- R → Ficheros generados en Racional
- E → Ejecutable



## Anexo 6

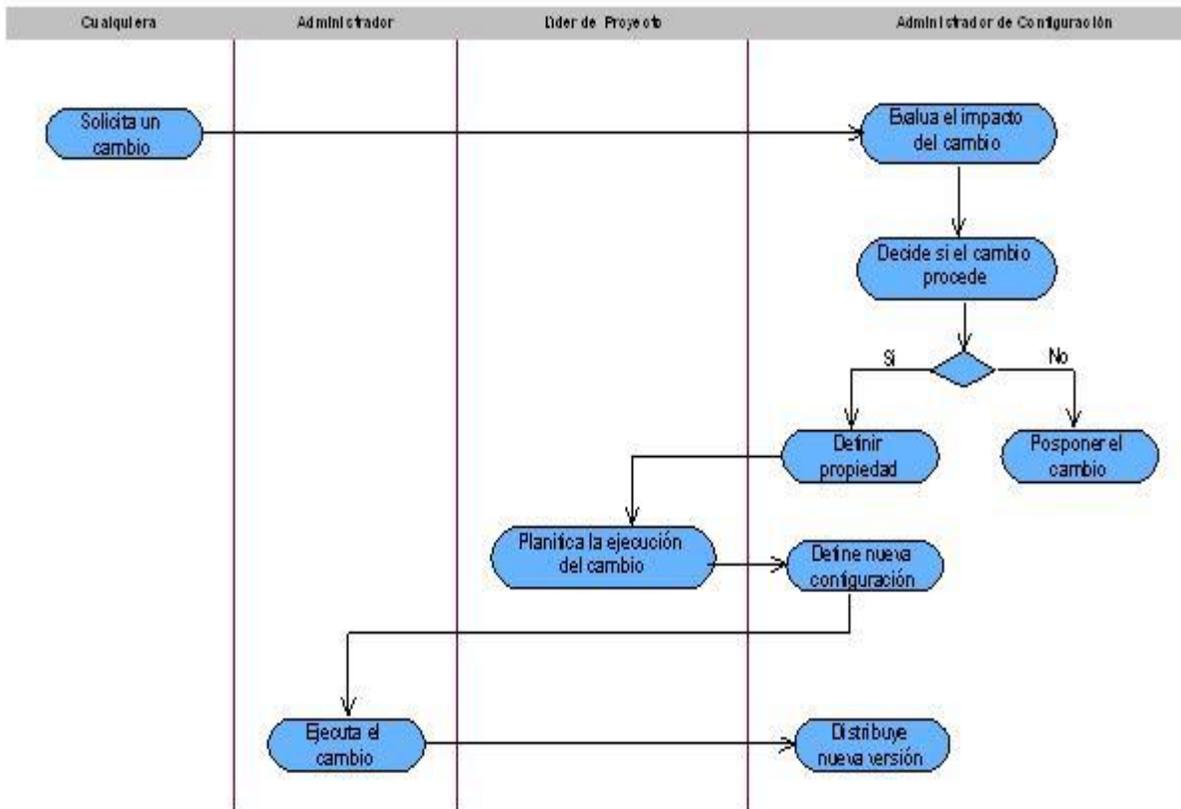
### Esquemas de control de cambio

Esquema de un proceso tradicional de control de cambios





Esquema moderno de un proceso de control de cambios





## Anexo 7

### Plantilla para documentar el Plan de Gestión de Configuración en el modelo ModelConfig.UCI

<Nombre del Proyecto>

Plan de Gestión de la Configuración

Versión <1.0>

---

#### Revisión Histórica

<b>Fecha</b>	<b>Versión</b>	<b>Descripción</b>	<b>Autor</b>
<dd/mm/yyyy>	<x.x>	<detalles>	<nombre>

---



## Tabla de Contenidos

1.	Introducción	4
1.1	Propósito	4
1.2	Alcance	4
1.3	Definiciones, Acrónimos y Abreviaturas	4
1.4	Referencias	4
1.5	Apreciación Global	4
2.	Gestión de la Configuración del Software	4
2.1	Organización, Responsabilidades e Interfaces	4
2.2	Herramientas, Ambientes e Infraestructura	4
3.	El Programa de CM	5
3.1	Identificación de la Configuración	5
3.1.1	Métodos de Identificación	5
3.1.2	Líneas Base del Proyecto	5
3.2	Control de los Cambios	5
3.2.1	Procesamiento y Aprobación de la solicitud de Cambio	5
3.3	Control de Versiones	5
3.3.1	Procesamiento y Aprobación de las Versiones	5
3.4	Contabilidad de Estado de la Configuración	5
3.4.1	Almacenamiento de los Medios del Proyecto y Proceso de Release	5
4.	Fechas Principales	6
5.	Entrenamiento y Recursos	6
6.	Control del Subcontratado y del Vendedor de Software	6

---

## Plan de Gestión de la Configuración

### Introducción

[La introducción del **Plan de Gestión de la Configuración** debe proporcionar una apreciación



global del documento entero. Debe incluir el propósito, el alcance, las definiciones, los acrónimos, las abreviaturas, las referencias utilizadas en el **Plan de Gestión de la Configuración.**]

### **Propósito**

[Especifique el propósito de este Plan de Gestión de la Configuración.]

### **Alcance**

[Una descripción breve del alcance de este **Plan de Gestión de la Configuración**; con qué modelo(s) está asociado, y cualquier cosa que es afectado o influenciado por este documento.]

### **Definiciones, Acrónimos y Abreviaturas**

[Debe proporcionar las definiciones de todos los términos, acrónimos, y abreviaturas requeridas para interpretar el **Plan de Gestión de la Configuración** apropiadamente. Esta información puede ser proporcionada por referencia al Glosario del proyecto.]

### **Referencias**

[Debe proporcionar una lista completa de todos los documentos referenciados en cualquier parte **Plan de Gestión de la Configuración**. Cada documento debe ser identificado por título, número del informe (si aplicable), fecha, y organización que publica. Especifique las fuentes de las que las referencias pueden obtenerse. Esta información puede ser proporcionada por referencia a un apéndice o a otro documento.]

### **Apreciación Global**

[Debe describir lo que el resto del **Plan de Gestión de la Configuración** contiene y explica cómo está organizado el documento.]

## **Gestión de la Configuración del Software**



### **Organización, Responsabilidades e Interfaces**

[Describa quién va a ser responsable de realizar las actividades de Gestión de Configuración (CM) descrito en el Procedimiento de CM.]

### **Herramientas, Ambientes e Infraestructura**

[Describa el ambiente de computación y las herramientas de software a ser utilizadas para cumplir las funciones de CM a lo largo del ciclo de vida del proyecto / producto.

Describa las herramientas y procedimientos requeridos para usar los elementos de configuración del control de versión generados a lo largo del ciclo de vida del proyecto / producto.

Elementos involucrados en establecer el ambiente de CM incluyen:

El tamaño anticipado de los datos del producto

La distribución del equipo

La ubicación física de las computadoras servidores y cliente]

### **El Programa de CM**

#### **Identificación de la Configuración**

Métodos de Identificación

[Describe cómo los artefactos del proyecto / producto serán nombrados, marcados y numerados. Ver Procedimiento de Identificación de los ECS.]

Líneas Base del Proyecto



[Las líneas base proporcionan una norma oficial en la que es basado el trabajo subsiguiente, y al cual sólo se hacen cambios autorizados.

Describe en qué puntos durante el ciclo de vida del proyecto / producto serán establecidas las líneas base. Las líneas base más comunes estarían al final de cada una de las fases de Concepción, Elaboración, Construcción y Transición. Las líneas base también podrían generarse al final de las iteraciones dentro de las varias fases o más frecuentemente aun.

Describe quién autorizará una línea base, y lo que va a ir en ella.]

### **Control de los Cambios**

Procesamiento y Aprobación de la solicitud de Cambio

[Selecciona procedimiento a seguir para el control de los cambios]

### **Control de Versiones**

Procesamiento y Aprobación de las Versiones

[Selecciona procedimiento a seguir para el control de versiones]

### **Contabilidad de Estado de la Configuración**

Almacenamiento de los Medios del Proyecto y Proceso de Release

[Describe políticas de retención, y planes de respaldo y recuperación. También describe cómo serán retenidos los medios (on-line, off-line, los tipos y el formato de los medios).

Se define:

Lugar físico de la Línea Base



Lugar físico del repositorio

Lugar físico de la biblioteca

### **Fechas Principales**

[Identifique las fechas principales internas y del cliente relacionadas con el esfuerzo de CM del proyecto / producto. Esta sección debe incluir detalles sobre cuando el propio CM Plan será actualizado.]

### **Entrenamiento y Recursos**

[Describa las herramientas del software, personal y entrenamiento requerido para llevar a cabo las actividades de CM especificadas.]

### **Control del Subcontratado y del Vendedor de Software**

[Describa como el software desarrollado fuera del ambiente del proyecto será incorporado.]

---



Anexo 8

Tabla de valores con que se analizó el coeficiente de los expertos a cada elemento evaluado

Fuentes de argumentación	Grado de influencia de cada una de las fuentes en su escritorio		
	Alto	Medio	Bajo
Estudios realizados por UD. relacionados con la GCS	0,5	0,4	0,2
Conocimientos que posee UD. relacionados con la InCuSoft	0,3	0,2	0,1
Conocimientos que posee UD. relacionados con Calidad de Software	0,3	0,2	0,1
Trabajos nacionales publicados por UD que guardan relación con el tema	0,14	0,10	0,06
Trabajos extranjeros publicados por UD., que guardan relación con el tema	0,08	0,06	0,04
Experiencia general en la temática	0,2	0,2	0,1
Posibilidad de decisión	0,5	0,4	0,3



## Glosario de Términos

**Automatizar:** Mecanizar ,motorizar.

**Calidad:** La calidad es un conjunto de características de una entidad que le confiere la aptitud para satisfacer las necesidades establecidas y las implícitas.

**Call Center:** Centro de llamadas

**Certificación:** Garantía que asegura la certeza o autenticidad de algo.

**Check in:** Chequee hacia dentro

**Check out:** Chequee hacia fuera

**Código:** Es la sucesión de sentencias que dan lugar al programa fuente.

**Componentes:** Que forma parte de alguna cosa o de su composición.

**Corporaciones:** Asociación u organismo oficial, generalmente público pero independiente de la administración estatal, con fines de utilidad pública

**Desk Help:** Ayuda del escritorio.

**Eclipse:** Es una plataforma de software de Código abierto independiente de una plataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido".

**Empresas:** Entidad integrada por el capital y el trabajo, como factores de la producción, y dedicada a actividades industriales, mercantiles o de prestación de servicios con fines lucrativos.

**Equipo de desarrollo:** Grupo de personas que se dedican a la realización de un producto determinado.



**Estadísticas:** Ciencia que utiliza conjuntos de datos numéricos para obtener inferencias basadas en el cálculo de probabilidades.

**Estándar:** Que tiene el tamaño, la forma o cualquier otra característica que sigue al modelo. Se aplica a lo que se produce en serie. Que sigue una tendencia muy extendida. Aquello que se considera modelo.

**Estratos:** División de la población en grupos

**Ficheros:** Consiste en una unidad de información en la que se almacena el resultado de utilizar un programa de proceso de textos, base de datos o cualquier otro tipo de aplicación. Necesita ser abierto mediante el programa que lo creó para poder operar con él.

**Globalización:** Integración de una serie de cosas en un planteamiento global.

**Grupos Focales:** Son por definición un acercamiento directo y cualitativo de una investigación. Es una entrevista conducida de una manera no estructurada y natural por un asesor entrenado entre un grupo pequeño de consumidores.

**Guía:** Lista de datos o información referentes a determinada materia.

**Hardware:** Conjunto de componentes materiales de un sistema informático. Cada una de las partes físicas que forman un ordenador, incluidos sus periféricos.

**Herramientas:** Aplicación empleada para la construcción (de ahí su nombre) de otros programas o aplicaciones.

**Hito:** Suceso o acontecimiento que sirve de punto de referencia.

**Implementar:** Implantar, poner en marcha una instalación informática.

**Iterar:** Repetir



**Línea Base:** Es un hito, un hecho memorable en el desarrollo de un software que es señalado por la aprobación de uno o más elementos de configuración, donde la aprobación de los mismos es a través de una Revisión Técnica Formal

**Método equiprobabilístico:** Se justifica porque procura estimaciones insesgadas de los parámetros de la población. La desviación de la equiprobabilidad aumenta los errores de la muestra.

**Métricas del software:** La continua aplicación de técnicas basadas en la medición al proceso de desarrollo de software y a sus productos para proveer información administrativa significativa y oportuna, junto con el uso de esas técnicas para mejorar el proceso y sus productos.

**Microsoft. Net:** Es un proyecto de Microsoft para crear una nueva plataforma de desarrollo de software con énfasis en transparencia de redes, con independencia de plataforma y que permita un rápido desarrollo de aplicaciones. Basado en esta plataforma, Microsoft intenta desarrollar una estrategia horizontal que integre todos sus productos, desde el Sistema Operativo hasta las herramientas de mercado.

**Modelo:** Cosa que ha de servir de objeto de imitación. Objeto, construcción u otra cosa con un diseño del que se reproduce más iguales. Esquema teórico de un sistema o de una realidad compleja que se elabora para facilitar su comprensión y estudio.

**Módulos:** Equipos de desarrollo.

**Muestreo:** Estudia las relaciones existentes entre una población y muestras extraídas de la misma.

**Muestreo aleatorio simple:** Una muestra aleatoria simple resulta cuando se seleccionan  $n$  elementos de una población, de manera que todas las combinaciones posibles de  $n$  elementos de la población tienen igual probabilidad de ser elegidas.



**Muestreo estratificado:** Las estimaciones de la población, basadas en la muestra estratificada, usualmente tienen mayor precisión (o menor error muestral) que si la población entera muestreada mediante muestreo aleatorio simple. El número de elementos seleccionado de cada estrato puede ser proporcional o desproporcional al tamaño del estrato en relación con la población.

**Muestreo polietápico o por conglomerado:** Una muestra de conglomerados, usualmente produce un mayor error muestral (por lo tanto, da menor precisión de las estimaciones acerca de la población) que una muestra aleatoria simple del mismo tamaño. Los elementos individuales dentro de cada "conglomerado" tienden usualmente a ser iguales.

**Muestreo sistemático:** Una muestra sistemática puede dar la misma precisión de estimación acerca de la población, que una muestra aleatoria simple cuando los elementos en la población están ordenados al azar.

**Normas:** La teoría de normalización consiste en obtener esquemas relacionales que cumplan unas determinadas condiciones y se centra en las determinadas Formas normales. Se dice que un esquema de relación está en una determinada forma normal si satisface un conjunto determinado de restricciones.

**On-line:** Indica que la aplicación o el sistema al que nos referimos permanece conectado a otro ordenador o a una red de ordenadores.

**Open Source:** Herramienta libre.

**Procedimiento:** Conjunto de instrucciones, controles, etc. que hacen posible la resolución de una cuestión específica. La impresión es un procedimiento, como lo es la incorporación de una imagen a un texto predeterminado, etc.

**Proceso:** Operación o conjunto combinado de operaciones con datos, o bien una secuencia de acontecimientos definida única y delimitada, que obedece a una intención operacional en condiciones predeterminadas. También se denomina proceso a una función que se está ejecutando.



**Producto:** Artefactos que se crean durante la vida del proyecto, como los modelos, código fuente, ejecutables, y documentación.

**Proyecto Productivo:** Elemento organizativo a través del cual se gestiona el desarrollo de software. El resultado de un proyecto es una versión de un producto.

**Reglas:** Son los métodos de producción de sentencias o instrucciones válidas que permitir y dan formar a un programa.

**Release:** Liberación de un producto.

**Requisitos Funcionales:** Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir.

**Rol:** Papel, cometido o función que tiene o desempeña que interpreta un actor.

**Sistema:** Conjunto de elementos que, ordenadamente relacionadas entre sí, contribuyen a determinado objeto.

**Software:** Término general que designa los diversos tipos de programas usados en computación.

**Soporte de Software:** El área de soporte de software es la encargada del estudio, desarrollo y programación de los distintos sistemas que se van a implementar.

**Subversion:** Es un software de sistema de control de versiones.

**Versión:** Una versión es una instancia de un elemento de Configuración. El término se usa para señalar a un elemento de Configuración del software que tiene un conjunto definido de características funcionales.

**Visual Source Safe:** Sistema de control de versiones de Microsoft para administrar el software y el desarrollo de sitios Web.